Q Light Controller Plus

User Documentation

Updated to version 4.5.1
November, 5th 2013

Basic Concepts & Glossary

Q Light Controller Plus (QLC+ for short) is meant to control lighting equipment used in various performances, like live concerts and theatres etc. The main purpose is that QLC+ could outperform commercial lighting desks with an intuitive and flexible user interface, without the need for a 500+ page manual.

This page has been arranged in alphabetical order to ease searching for a specific topic.

Audio

An audio function is an object representing an audio file stored in a disk.
Audio files can be placed in a show at the desired time, using the Show Manager panel.

Blackout

Blackout is used to set all channels in all universes to zero and keep them that way, regardless of what functions are currently being run or what values have been set to them manually. When blackout is turned off, current values of all channels are set to each universe.

Capabilities

Some channels in intelligent fixtures provide many kinds of functions, or *capabilities* like switching the lamp on when the channel value is [240-255], setting a red color on a color wheel when the value is exactly [15], or simply controlling the fixture's dimmer intensity with values [0-255]. Each of these individual function is called a capability and each of them has these three properties:

- Minimum value: The minimum channel value that provides a capability.
- Maximum value: The maximum channel value that provides a capability.
- Name: The friendly name of a capability.

Channel Groups

Channel groups can be added and defined in the Fixture Manager panel by using the Channel groups editor.
Channel groups can have a user defined name and can group any user defined channels selected from an existing Fixtures list.

Chaser

A chaser function encapsulates multiple scenes that are run in sequence, one after the other, when the chaser function is started. The next function is run only after the previous has finished. Any number of functions can be inserted to a chaser.

Chaser functions' direction can be reversed. The function can also be set to do an infinite loop, an infinite ping-pong-loop (direction is reversed after each pass) or it can run thru just once, in a single-shot mode, after which it terminates by itself. If the function is set to do an infinite loop, it must be stopped manually.

As of version 3.3.0, each chaser has their own speed settings:

- **Fade In:** The fade in speed of a step
- **Hold:** The hold time of a step
- **Fade Out:** The fade out speed of a step
- **Duration:** The duration of a step

Copies of chaser functions can be created with the Function Manager. The scenes inside a chaser are not duplicated when a chaser is copied. Only the order and direction are copied to the new one.

Click And Go

Starting from version 4.3.0, QLC+ can benefit of the Click And Go technology. This set of functionalities will allow users to quick access macros and colors in a completely visual way and with just a couple of clicks. This can lead to more efficient live shows and more freedom to choose the desired result very easily.
So far, three types of widgets are available:

- Single Color (applies to: Red, Green, Blue, Cyan, Yellow, Magenta and White intensity channels)
- RGB Color Picker. Controls values for selected RGB channels with a single click
- Gobo/Macro Picker. Access and display a Gobo/Macro defined in the Fixture definition

An overview with screenshots is available here

Collection

A collection function encapsulates multiple functions that are run simultaneously when the collection function is executed. Any number of functions can be inserted to a collection, but each function can be inserted only once and a collection cannot be a direct member of itself.

Collections have no speed settings. Each member function's speed is set individually thru their own editors.

Copies of collection functions can be created with the Function Manager. The functions inside a collection are not duplicated; only the list of functions is copied.

DMX

DMX is short for **D**igital **M**ultiple**X**. It basically defines a whole bunch of properties, protocol, wiring etc. but in the case of a lighting software, it defines the maximum number of channels (512) per universe and the channels' value range (0-255).

QLC+ supports 4 individual universes, that need not be necessarily connected to DMX hardware; rather, DMX has just been selected as the de facto lighting standard. Actual hardware abstraction (whether it's analog 0-10V, DMX or some other method) is achieved thru output plugins.

EFX

An EFX function is mainly used to automate moving lights (e.g. scanners & moving heads). The EFX can create complex mathematical paths on an X-Y plane that are converted to DMX values for the fixture's pan and tilt channels. **Only fixtures that contain valid pan & tilt channels can take part in an EFX function.**

As of version 3.3.0, each EFX has their own speed settings:

- **Fade In:** The time it takes for the fixtures in the EFX to fade their intensity channels up
- **Fade Out:** The time it takes for the fixtures in the EFX to fade their intensity channels back to zero
- **Duration:** The duration of one full round using the selected pattern

Fixtures taking part in an EFX function can be set to follow the algorithm in certain order:

- **Parallel:** all fixtures follow the same pattern synchronously
- **Serial:** fixtures start following the pattern one after the other, with a little delay between each of them.
- **Asymmetric:** all fixtures start moving simultaneously, but with similar offset as in the Serial mode.

EFX functions' direction can be reversed for all fixtures at once or on a per-fixture basis. The function can also be set to do an infinte loop, an infinite ping-pong-loop (direction is reversed after each pass) or it can run thru just once, in a single-shot mode, after which it terminates by itself. If the function is set to do an infinite loop, it must be stopped manually.

Copies of EFX functions can be created with the Function Manager. All of the EFX's contents are copied to the duplicate.

Fixtures

A fixture is essentially one lighting device. It can be, for example, one moving head, one scanner, one laser etc.. However, for simplicity, individual PAR cans (and the like) that are usually controlled thru one dimmer channel per can, can be grouped together to form one single fixture.

With the Fixture Definition Editor, users can edit shared fixture information stored in a fixture library that contains the following properties for each fixture:

- Manufacturer (e.g. Martin)
- Model (e.g. MAC250)
- Type (Color Changer, Scanner, Moving Head, Smoke, Haze, Fan...)
- Physical properties (bulb type, beam angle, dimensions...)
- Channels:
    - Channel group (Intensity, Pan, Tilt, Gobo, Color, Speed etc.)
    - 8bit and 16bit channel bindings for pan & tilt groups
    - Optional primary colour for intensity channels (RGB/CMY)
    - Value ranges for channel features (e.g. 0-5:Lamp on, 6-15:Strobe etc..)

These fixture definitions can then be used to create actual fixtures, that will in the Q Light Controller Plus application, have additional properties defined by users:

- DMX Universe
- DMX Address
- Name

Several instances of a fixture can be created (i.e. users must be able to have several instances of a MAC250 in use). Each fixture can be named, but the name is not used internally to identify individual fixture instances. The same goes for the DMX address. Nevertheless users are encouraged to name their fixtures in some systematic way to help identify each of them -- if necessary.

Generic dimmer devices don't need their own fixture definitions, because usually multiple dimmers are patched into a common address space, employing one or more dimmer racks. Users can create instances of these generic dimmer entities just by defining the number of channels each one of them should have.

Fixture Group

A fixture group is, as the name says, a group of fixtures. They also define (in a rather basic level) the actual physical, real world arrangement of these fixtures. This knowledge can be used for example in the RGB Matrix to produce a wall of RGB-mixable lights that can act as individual pixels in a graphic pattern or scrolling text.

Fixture Mode

Many manufacturers design their intelligent lights in such a way that they can be configured to understand different sets of channels. For example, a scanner might have two configuration options: one for only 8bit movement channels (1x pan, 1x tilt) and another one for 16bit movement channels (2x pan, 2x tilt). Instead of creating a completely new fixture definition for each variation, they have been bundled together in QLC+'s fixture definitions into fixture modes.

Functions

The number of functions is practically unlimited. Functions are used to automate the setting of values to DMX channels. Each function type has its own way of automating lights.

The function types are:

- Scene
- Chaser
- Sequence
- EFX
- RGB Matrix
- Collection
- Show
- Audio

Each function can be named and although the name is not used to uniquely identify individual functions, users are encouraged to name their functions in some systematic and concise way to help identify each of them. For your own comfort.

As of version 3.3.0, each function has their own speed settings:

- **Fade In:** The time used to fade HTP (in Scenes also LTP) channels to their target value
- **Fade Out:** The time used to fade HTP/intensity channels back to zero
- **Duration:** The duration of the current step (not applicable on Scenes)

Grand Master

The Grand Master is used as the final master slider before values are written to the actual physical DMX hardware. Usually, the Grand Master affects only **Intensity** channels, but can be also changed to have effect on the values of **all** channels.

The Grand Master has also two **Value Modes** that control the way *how* the Grand Master affects channel values passing through:

- Reduce: Affected channels' values are reduced by a percentage set with the Grand Master slider. For example, Grand Master at 50% will result in all affected channels being reduced to 50% of their **current** values.

- Limit: Affected channels cannot get larger values than directly set with the Grand Master slider. For example, Grand Master at 127 will result in all affected channel's maximum values being limited at exactly 127.

Head

A head represents an individual light output device in a fixture. Usually, a single fixture contains exactly one output, like the lens, the bulb, or a set of LEDs. There is, however, an increasing amount of fixtures in the market that are treated as a single fixture, but they have multiple light output devices, i.e. heads.

For example, you might have a RGB LED bar fixture that is packed into a single chassis, and as such it appears as a single fixture with one DMX input and one DMX output, but that actually comprises of four separate RGB LED "fixtures". These separate fixtures are treated in QLC+ as heads; they share some common properties with their sibling heads, they can be controlled individually, but they might also have a master intensity control that controls the light output of all the heads together.

Each head belongs to a Fixture Mode because in one mode, a fixture might provide enough channels to control each of its heads individually while in another mode, only a handful of channels might be provided for controlling all heads simultaneously.

HTP (Highest Takes Precedence)

All channels that have been assigned to the **Intensity** group obey the very simple HTP rule: The highest value (nearer 100%) that is currently set to the channel is the one that gets sent out to the DMX universe. This is particularly useful for dimmer channels that namely control the *light intensity*. Let's say you have two sliders that control the same intensity channel. First, you set slider 1 to 50% and then slider 2 to 75%; as long as slider 2 is below 50%, nothing happens but after crossing the 50% value set by slider 1, the light intensity increased all the way to 75%. If you drag slider 2 again to 0%, the light intensity diminishes until it reaches the 50% set by slider 1 and stays there until slider 1 is dragged down. See also LTP.

Input plugins

Primary input methods for QLC+ are naturally the keyboard and mouse. Users can attach keyboard keys to virtual console buttons and drag sliders and do pretty much everything with a mouse. It is possible to attach additional input devices to one's computer to alleviate the rather clumsy and slow user experience that is achieved with a regular mouse and a keyboard. Special input plugins provide capabilities for getting external devices (mainly slider boards) to produce input data to various QLC+ elements.

Input (line)

An input is a physical input line provided by some hardware which is accessed thru an input plugin. It can be, for example, a MIDI IN connector in the user's computer (or peripheral) to which users can connect MIDI-capable input devices like slider boards etc.

Input profiles

Input profiles can be thought of as fixtures' cousins; they contain information on specific devices that produce input data. An input device can be, for example, a slider board like the Behringer BCF-2000, Korg nanoKONTROL, an Enttec Playback Wing...

LTP (Latest Takes Precedence)

All channels that have been assigned to the **Intensity** group obey the very simple LTP rule: The latest value that was set by a function or a slider gets sent out to the DMX universe. This scheme is useful for *intelligent fixture parameters* not related to *intensity*. For example, a gobo change is expected to happen immediately, regardless of what some other control has done to it. See also HTP.

Modes

Q Light Controller Plus is based on the common concept of having two distinct operational modes to prevent accidental or otherwise harmful changes during operation:

- Design mode is meant to edit the behaviour of the program; create and edit functions and fixtures and how they work.
- Operate mode is meant to execute the created functions that eventually control the user's lighting fixtures.

Output plugins

DMX output is achieved thru output plugins that can support one or more output lines. It is also possible to run QLC+ without actual DMX hardware.

Output (line)

An output is a physical output line provided by a hardware device which is accessed thru an output plugin. In other words, it is a real DMX universe, but has been dubbed output to separate it from QLC+'s internal universes. You could think of them as individual XLR output connectors in your DMX hardware.

RGB Matrix

An RGB matrix function can be used to impose simple graphics and text on a matrix (a grid or a wall) of RGB and/or monochrome fixture heads. The RGB matrix function has been designed to be extendable with scripts that can be written by users.

Each RGB matrix has its own speed settings:

- **Fade In:** Time to fade each pixel ON
- **Fade Out:** Time to fade each pixel OFF
- **Duration:** The duration of the current step/frame

RGB Script

An RGB script is a program written in ECMAScript (also known as JavaScript) that produces the necessary image data for RGB Matrix functions. Learn more from the RGB Script API page.

Scene

A scene function encapsulates the values of selected channels owned by one or more fixture instances. When a scene is started, the time it takes for its channels to reach their target values depends on the function's speed settings:

As of version 3.3.0, each function has their own speed settings:

- **Fade In:** The time used to fade all channels to their target values, from whatever value they had
- **Fade Out:** The time used to fade HTP/intensity channels back to zero. Note that ONLY HTP channels are affected by this setting.

Copies of scene functions can be created with the Function Manager. All of the scene's contents are copied to the duplicate.

Sequence

A sequence is a special Chaser which is bound to a specific Scene. It means all the created steps will control channels of the same scene. When creating new steps in a sequence, no Function selection popup will appear, since a Sequence is a child of a Scene.
A sequence will appear in the Function Manager with a different icon than a Chaser.

Show

A Show is an advanced function which encapsulates scenes, audio and sequences to create a time driven light show. A Show can be created only with the Show Manager and can be inspected and renamed with the Show Editor.

Questions & Answers

In this page you will find the common questions that may come to your mind when starting with QLC+.
Here you can find the direct answer or a help to point you in the right direction.

Q: QLC+ cannot detect my USB device

A: QLC+ supports a wide variety of USB devices. First of all you should check if the physical connection is OK. Usually a LED on your device should indicate if it is powered up and working correctly.
If you are using Windows and your device is manufactured by Peperoni or Velleman, please read the information on how to get them working directly on these help pages. For licensing issues they both need an extra DLL file to work. Please check Peperoni output plugin or Velleman output plugin
If you're using Linux, please check if your distribution detected the device when plugged in. Basically the "dmesg" command should tell you something.

Q: I've got several buttons in my Virtual Console. I need a way to disable the currently active button when I enable another one. How do I do that ?

A: Simply place your buttons inside a Solo Frame. It does exactly that.

Q: When I start EFX, all fixtures are at full intensity and I can't make them dimmer.

A: See this note.

Q: I just upgraded my Mac to OSX Mavericks but my Open USB DMX adapter doesn't work anymore.

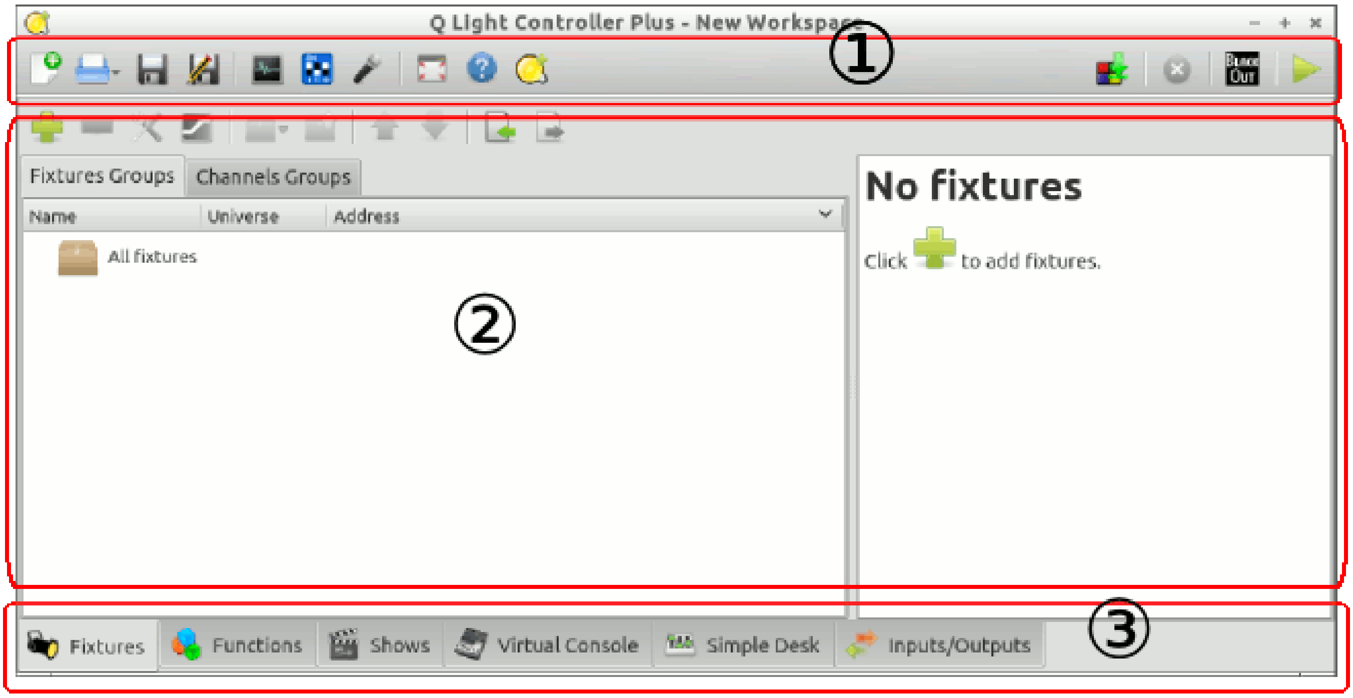
A: The problem is in a new Apple USB driver. Following the procedure to disable it.
Please note that this can compromise the behaviour of other USB devices, so do it only if you know what you're doing !
From a terminal type the following commands:

```
cd /System/Library/Extensions/IOUSBFamily.kext/Contents/PlugIns
sudo mv AppleUSBFTDI.kext AppleUSBFTDI.disabled
sudo touch /System/Library/Extensions
```

Main Window

QLC+ Main consists of three main parts:

1. Menu bar contains buttons for global functions
2. Active panel selected by respective tab
3. Tabs allow selecting one of the QLC+ panels



Most controls throughout the software have tooltips, which pop up after holding your mouse cursor over them for a while.

Menu bar (1)

The menu bar on the top of the workspace window contains the following buttons (from left to right):

New workspace

Open an existing workspace (hold the button longer for a list of recent files)

Save the current workspace

Save the current workspace with a new name

DMX Monitor

DMX Address tool

Audio triggers

Toggle full screen mode

View this documentation

View information about QLC+

DMX Dump

Live Edit: allows you to modify a function while QLC+ is in Operate Mode

Stop All Functions

BLACK OUT Toggle Blackout

Switch between Design Mode and Operate Mode

Active Panel (2)

Everything happens here. What panel is active depends what tab is currently selected.

Panel Tabs (3)

At the bottom of main window you can find how to switch between QLC+ panels with easy eye catching icons that appear as follows (from left to right):

View the Fixture Manager

View the Function Manager

View the Show Manager

View the Virtual Console

View the Simple Desk

View the Input/Output configuration manager

DMX Monitor

The DMX Monitor displays the current values that are being sent to the output universes. Only those channels are displayed that have actual fixtures behind them. The monitor's display options have no effect on actual fixture addressing, after all, it is just a **monitor**.

Controls

Aa	Change the monitor font. To prevent the numbers from flickering and jumping, you should choose a proportional (i.e. fixed width) font; for example Monaco, Andale or Courier.
DMX Channels	Display fixtures' channel numbers as absolute DMX channels; channel numbers go from 1 to 512 as they are assigned to each fixture.
Relative Channels	Display fixtures' channel numbers relative to fixtures i.e. every fixture's channel numbers always start from 1.
DMX Values	Display channel values as absolute DMX values (0-255).
Percent Values	Display channel values as percentages of 255 (0-100%).

DMX Address Tool

The DMX address tool is a quite easy/self explanatory functionality of QLC+ introduced in version 4.3.2.
It helps you to quickly calculate your fixtures addresses in a visual way.
A representation of a common 10pin DIP switch is displayed and you can select the background color, the horizontal and vertical oriantation, to display the DIP exactly how you see it on your fixture.

A particular note goes for the 10th pin. Usually it is used to enable/disable the DMX feature of a fixture so, if your product manual says it has to be raised to enable the DMX control, then remember to do it even if QLC+ displays it set to OFF.

DMX Dump

The DMX Dump functionality allows you to save the current DMX values that are sent to the ouput universes in a particular moment. Basically it makes a "snapshot" of DMX channels and saves them for a later use.
DMX Dump can save values to a new Scene and also add the newly created Scene into an existing Chaser, Virtual Console button or slider

Please note that:

- Selected Channels/Chasers will be remembered when re-opening this window
- The DMX values saved are taken before the Grand Master

When opening the DMX Dump window, the following options will appear:

Scene name	Defines the name that the Scene that will be created will have. If none specified, a default name like "New Scene From Live" and a numeric identification will be set for fast use of this functionality.
Dump all channels	If this option is selected, QLC+ will dump all the channels of all the universes and all the fixtures. To inform the user about what this option will do, a report in the form of (Universes, Fixtures, Channels) will be displayed.
Dump selected channels	If this option is selected, the panel below will be activated, allowing you to choose exactly the channels that you need to be saved into a Scene. They're organized in a tree by Universes and Fixtures.
Add to	This section offers you to immediately use your scene into and existing Virtual Console widget. Options are: • **Chaser**: The list is filled with all the chasers present when DMX Dump window is opened. Each Chaser has a checkbox that, if checked, will tell QLC+ to add the newly created Scene to the selected chasers. This feature is nice when using Cue Lists in the Virtual Console panel. In this case, the newly created Scene will appear in the Cue List for immediate use during a live performance. • **Button**: The list is filled with all the buttons currently present in your Virtual Console space. When selected, the buttons will be set to activate/deactivate the Scene just captured. You will see the button label chaning to "Scene from live ..." and a progress number to distinguish it. **Note:** Any previous function associated to the selected buttons will be overwritten ! • **Slider**: The list is filled with all the sliders currently present in your Virtual Console space. Similarly to buttons, all the selected sliders will be set to control the Scene just captured. **Note 1:** A slider must be in **playback mode** to work as an intensity controller for a Scene. **Note 2:** Any previous function associated to the selected sliders will be overwritten !

Dump only non-zero values	This option will tell QLC+ to save only the channels with values **not** equal to zero. If you know what you're doing, this could save project space and avoid channels conflicts with other Virtual Console widgets.

Live Edit

Starting from version 4.5.0, QLC+ offers a functionality to adjust your Functions while in Operate Mode.

The Live Edit icon is located in the QLC+ top bar, beside the DMX Dump icon and it is activated only when the user switch to Operate Mode.
When clicking on the Live Edit icon, a Function selection panel will be displayed, allowing to select the function you are willing to adjust.
When pressing OK, the proper editor will be displayed for function editing.
At the moment the following functions are supported for live editing:

- Scene will open a Scene Editor
  Note that by default the editor will be set in "Blind mode" in case you want to make the changes effective only the next time the Scene is being played.
- Chaser will open a Chaser Editor
- EFX will open a EFX Editor
- RGB Matrix will open a RGB Matrix Editor

Fixture Manager

The fixture manager is the heart of QLC+'s fixture-oriented architecture. As its name already says, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left side of the manager there is a list that displays all of the fixtures in the current workspace. When a fixture is selected, the right side of the window displays the details of the selected fixture. When a Fixture Group is selected, the right side of the window is occupied by the Fixture Group Editor.

Controls

✚	Add new Fixture(s) to the workspace with the Add/Edit Fixture dialog.
▬	Remove the selected fixtures from the workspace. This also removes the fixtures from ALL groups they have been assigned to.
🛠	Edit the currently selected fixture's properties with the Add/Edit Fixture dialog.
⟋	Opens a window to select the desired channels to exclude from fade transitions. For example, moving head motors shouldn't be affected from Fade In/Fade Out curves. With this option you can exclude them to obtain the correct behaviour. When dealing with a lot of fixtures of the same type, you might want to check the "Apply changes to fixtures of the same type" check box. This can save you quite a lot of time when setting up your projects for the first time.
📦	Assign the selected fixtures to a Fixture Group displayed in a popup menu. If you have no groups defined yet, you can choose to create a new one from the menu that opens when this button is clicked.
📤	Resign (remove) the selected fixtures from the group they are currently in. Note that removing fixtures from a group will NOT destroy the fixtures completely. Also, removal from one group will not affect the fixtures' memberships to other groups.
⬆	Move the selected Channel Group up
⬇	Move the selected Channel Group down
📥	Available from version 4.2.0. Import a fixture list file (.qxfl) into QLC+. Please note that fixture addresses conflicts are not handled, so it is suggested to use this functionality on an empty project.
➡	Available from version 4.2.0. Export the list of fixtures currently available on a QLC+ project into a file with extension .qxfl. This file can be used afterward with the import functionality.
🔀	Available from version 4.4.1. Open the Fixtures remapping window.

Add/Edit Fixtures

The same dialog (with slight differences) is used for both adding new and editing the properties of existing Fixtures.

NOTE: When editing an existing fixture, you can always choose a completely different fixture type to replace the previous one, but if their channels don't match each other exactly, any Scene functions you may have created, will very probably do things that you don't want. Also, if the channel counts don't match, you might end up having fixtures with overlapping channels, which also results in unwanted behaviour. RGB Matrix, EFX, Chaser and Collection functions remain unaffected since they don't directly address specific channels, but are more dynamic in nature.

Controls

Fixture Model list	Shows you a list of available fixture definitions and their general types. If you don't see your favourite fixture definition on the list, you can create one yourself with the accompanying, easy to use **QLC+ Fixture Editor**. Since QLC+ is completely free software, please consider sharing the definition files with the community at https://sourceforge.net/p/qlcplus/discussion/fixtures/
Fixture Properties	• Name: The friendly name you wish to assign to the fixture • Mode: The mode you have configured for the fixture • Universe: The DMX universe that the newly-added fixture(s) are connected to • Address: The address of the (first) fixture you are adding. If you are adding multiple fixtures, each consecutive fixture will be assigned an address immediately after the previous fixture's channels, unless the address gap (see below) is non-zero. • Channels: If you have selected the **Generic** dimmer device, you can define its channel count in this field. Otherwise this field is read-only and it just tells you how many channels the currently selected fixture needs in its currently selected mode. • Channel list: This field displays a more detailed list of individual channels for the currently selected fixture and its mode. For **Generic** dimmers, this field remains empty since all dimmer channels are treated as dummy **intensity** channels.
Multiple Fixtures	You can also add more than one fixture at the same time if your setup consists of multiple fixtures that are of the same make and model. These features are disabled when you are editing an existing fixture. • Amount: The number of new fixtures to create. Each newly-added fixture also includes a number after its assigned name when you add more than one fixture at a time. • Address gap: Leave this many empty channels between each new fixture

Fixture Group Editor

The Fixture Group Editor is used when you select a Fixture Group in the Fixture Manager. This editor can be used to describe the physical arrangement of the fixtures and their heads that are assigned to the group.

Controls

Each Fixture Group has a name (for your convenience) and the fixture configuration in an unlimited [X,Y] grid. You can adjust the size of the grid as well as the placement of individual fixture heads on the grid.

Fixture Group Name	Change the name of the group.
➡	Add fixture heads to the currently selected row, filling each subsequent column on that row with the selected fixture heads. You can select individual fixture heads or complete fixtures to be added on the selected row.
⬇	Add fixture heads to the currently selected column, filling each subsequent row on that column with the selected fixture heads. You can select individual fixture heads or complete fixtures to be added on the selected column.
▬	Remove the selected fixture head from the grid (and the group).
Fixture grid	The fixture grid displays the current fixture/head arrangement in the selected group. You can switch places between two heads as well as fill up spaces or create empty spaces between heads simply by dragging the heads on top of each other in the grid. When using fixture groups in RGB Matrixes, each cell in the grid represents one colored (or monochrome) pixel in a graphic scene rendered by the RGB Matrix. Each cell is displayed with an info text that contains the following information: • The name of the fixture as well as an icon • **H:** represents the head number • **A:** represents the fixture's address • **U:** represents the fixture's universe
Width	Adjusts the grid width (number of columns on the X-axis).
Height	Adjusts the grid height (number of rows on the Y-axis).

Channel Groups Editor

The Channel Groups editor it activated when you click on the "Channels Groups" tab in the Fixture Manager panel.
With this functionality (introduced in QLC+ version 4.0.0), it is possible to create groups of channels with the same functionality.
For example if you have 20 PARs you might want to control the RED channel of all of them with a single fader.

Controls

➕	Add a new channels group to the workspace with the Add/Edit Channels Group dialog.
➖	Remove the selected channels group from the workspace.
🛠	Edit an existing channels group in the workspace with the Add/Edit Channels Group dialog.
⬆	Move the selected group up in the list to change the logical order.
⬇	Move the selected group down in the list to change the logical order.

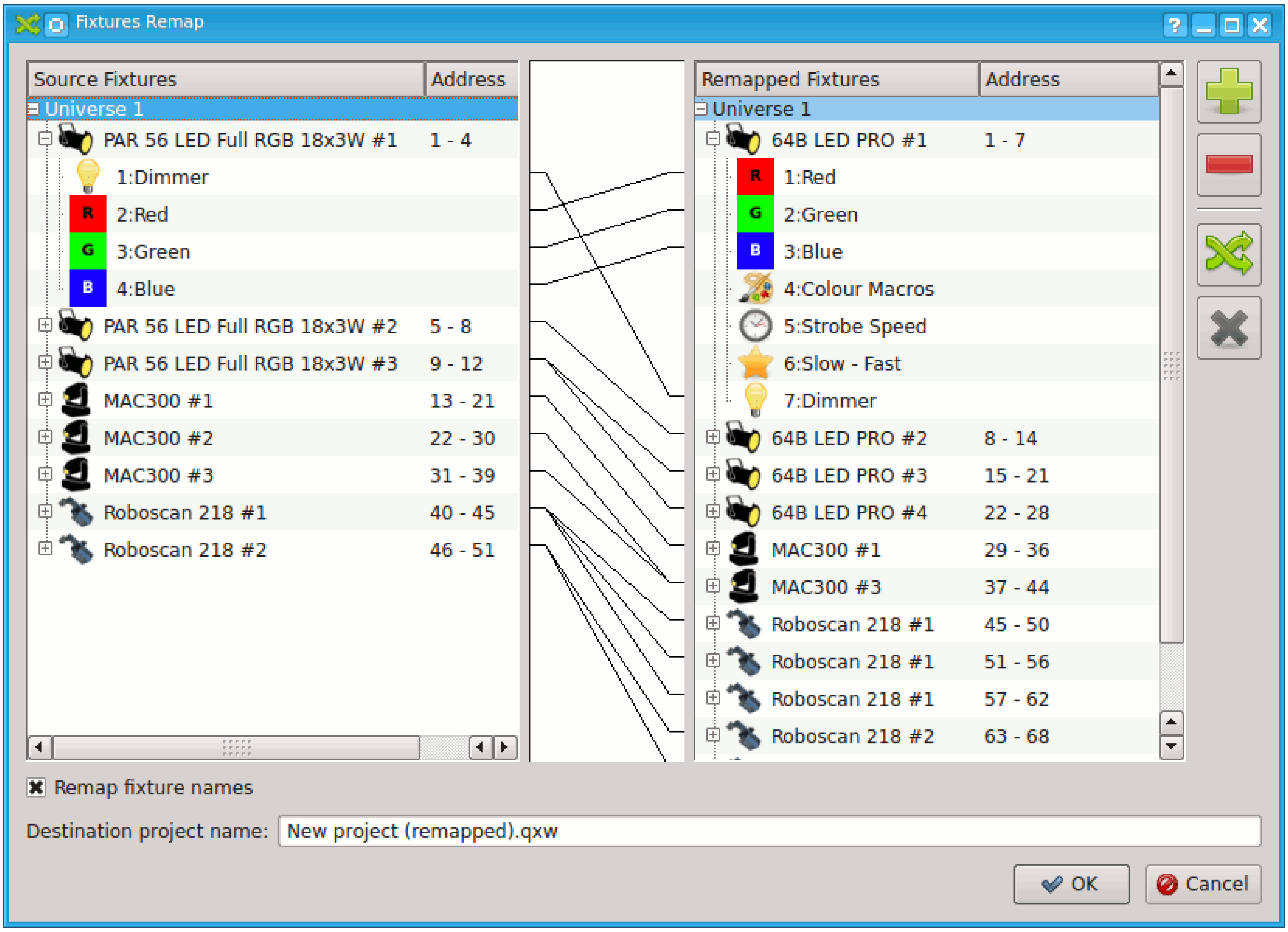
Fixtures remapping

Starting from version 4.4.1, QLC+ offers a functionality called fixtures remapping.
When performing live shows in different venues, you may only be able to find out at the last minute which fixtures are installed there. Well, fixtures remapping helps you to use your existing projects in this and many other situations, such as when you need to replace a faulty fixture or when you want to use hired in equipment alongside your own.
For example, you can set up a project with just one PAR, one moving head and one scanner. When reaching the venue where the show is going to take place, you can remap your fixtures to those you find there, for example 50 PARs, 30 moving heads and 15 scanners.
With QLC+, it takes just a few minutes to do this operation !

Fixtures remapping allows you to perform 1-to-1 or 1-to-many reassignments of entire fixtures or single channels. QLC+ will try as far as possible to reassign the original channels used in the project to new channels in the same category.
When you confirm the remapping operation, a new project will be automatically saved to preserve your original project.
All the fixtures, scenes, EFX, Virtual Console widgets and audio triggers found in the original project will be remapped so that they work on the new fixtures list.

Remapping window

Let's now explain how to use the fixture remapping window, starting from a screenshot of a complete example:



Source Fixtures	Address
Universe 1	
PAR 56 LED Full RGB 18x3W #1	1 - 4
1:Dimmer	
2:Red	
3:Green	
4:Blue	
PAR 56 LED Full RGB 18x3W #2	5 - 8
PAR 56 LED Full RGB 18x3W #3	9 - 12
MAC300 #1	13 - 21
MAC300 #2	22 - 30
MAC300 #3	31 - 39
Roboscan 218 #1	40 - 45
Roboscan 218 #2	46 - 51

Remapped Fixtures	Address
Universe 1	
64B LED PRO #1	1 - 7
1:Red	
2:Green	
3:Blue	
4:Colour Macros	
5:Strobe Speed	
6:Slow - Fast	
7:Dimmer	
64B LED PRO #2	8 - 14
64B LED PRO #3	15 - 21
64B LED PRO #4	22 - 28
MAC300 #1	29 - 36
MAC300 #3	37 - 44
Roboscan 218 #1	45 - 50
Roboscan 218 #1	51 - 56
Roboscan 218 #1	57 - 62
Roboscan 218 #2	63 - 68

☒ Remap fixture names

Destination project name: New project (remapped).qxw

And now, here is a detailed explanation of each element of the remapping window.

Source fixtures	On the left side of the window, there is a tree representing the universes, fixtures and channels used in your project. This list cannot be changed in this

fixtures	window.
Remapped fixtures	On the right side of the window, there is a tree representing the universes, fixtures and channels where you are going to remap your source fixtures.
✚	Add one or more fixtures to the Remapped fixtures list. Clicking this button will open the Add/Edit Fixtures window. Please note that once a fixture is added, it cannot be modified, so if you need specific names or quantity, remember to do it before adding the new fixtures
▬	Remove the selected Fixture from the remapped fixtures list.
🔀	This is probably the most important button in the window. It allows you to determine the connection between a source ficture and a remapped fixture. The connections can be performed either between single channels or on whole fixtures. In the first case you will need to select the source channel from the source fixtures list and a target channel from the remapped fixtures list. In the second case you will need to select a fixture from the source fixtures list and a target fixture from the remapped fixtures list. Wrong selections will cause an error message to popup. For example you cannot remap a channel to a fixture and vice-versa. If the connection is valid, it will be represented as a line in the white area between the source list and the remapped list.
✖	Remove a previously created remapping connection. Wrong selections will cause an error popup.
Remap fixtures names	If this is checked, a new remapping connection between fixtures will also set the target fixture name be the same as the source fixture name.
Destination project name	The absolute path and name of the remapped project. For convenience, QLC+ will automatically take the original project name and will add "(remapped)" at the end of it.

Adding Fixtures

Fixture Manager

Open the fixture manager by clicking its button on the main toolbar. Alternatively you can select Fixtures from the Manager menu.

The fixture manager is the heart of QLC+ fixture-oriented architecture. As its name already says, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left side of the manager there is a list that contains all of the fixtures in the current workspace (it will be empty until you have added some fixtures). On the right side you can see some common information related to the currently selected fixture. At the top of the fixture manager there is again another toolbar, containing the following buttons (from left to right):

Add new fixtures

Remove selected fixtures

Configure the selected fixture

Group a fixture selection

Ungroup a fixture from a group

Import a previously saved list of fixtures

Export a list of fixtures

Adding a fixture

Add a fixture to the workspace by clicking the add button.

On the left side of the dialog you can see a list of available fixture manufacturers. Each manufacturer item is actually a folder containing a number of fixture models produced by the manufacturer. You can find for example a "DJScan250" under the "Futurelight" folder. As you click a fixture from the list, you can see the Channels field on the right side change to display the number of DMX channels required by the selected fixture. There is also a list of the fixture's channels just under the Channels box.

You can edit the new fixture's name in the Name field or you can stick to the default that the application suggests. If the fixture has different operational modes (different sets of channels), you can select one from the Mode box. The fixture's DMX address can be set in the Address field and should be the same as the actual physical fixture's DMX address. The Universe field is used to assign the fixture to a physical DMX output universe. Usually each universe has its own cable coming from the computer.

If you wish to add multiple fixtures of the same type, you can increase the value in the Amount box. If you wish to leave a gap between each fixture's address space, you can change the value in the Address gap box.

If you don't understand the DMX addressing principles, please consult your lighting equipment manuals for more information. In short, a DMX address is the first DMX channel of a fixture. In the case of a DJScan250 (which uses 6 channels), assigning for example 1 as its DMX address, reserves channels 1, 2, 3, 4, 5, and 6 for the fixture. The next fixture must then be assigned to DMX address 7 to prevent channel overlapping.

Adding a generic dimmer

Dimmers are somewhat special devices, since all they can usually do is to adjust the intensities of their channels (which usually drive PAR cans and the like). If you wish to add such a device, select the Generic dimmer fixture from the list and specify the number of channels you wish the device to employ in the Channels box. Note that you cannot adjust the mode for dimmers and the channel list stays empty all the time.

Back to the fixture manager

Click OK to close the dialog and add the selected fixture(s) to the workspace.

On the left side of the fixture manager you can now see the fixture(s) that you just added. On the right side, you can see information on the currently selected fixture. You can edit the fixture's name, address and universe by clicking the Configure button. You can also change the fixture definition by using the configuration dialog.

Function Manager

With the function manager you can manage all of your functions that do the actual work to automate your lights. You can create new functions, remove and edit existing ones as well as create copies of them. Controls are on the upper part of the window; the lower part displays all of your functions as well as the function editor for the currently selected function (if any).

Each function type can be found under its own category: Scene for scenes, EFX for EFX's etc. When a new function is created or an existing one selected, the appropriate editor is chosen and displayed on the right side of the Function Manager window. Changes made in the editor pane are stored immediately to the functions themselves and no additional OK clicks are no longer required.

Controls

▦	Create a new Scene and edit it using the Scene Editor.
➡	Create a new Chaser and edit it using the Chaser Editor.
●●●	Create a new Sequence and edit it using the Chaser Editor.
💡	Create a new EFX and edit it using the EFX Editor.
▤	Create a new Collection and edit it using the Collection Editor.
●	Create a new RGB Matrix and edit it using the RGB Matrix Editor.
🎞	Create a new RGB Script and edit it using the RGB Script Editor.
♫	Create a new Audio function to be used in Chasers or Shows.
🪄	Start the Function Wizard.
⧉	Create a copy of each of the selected functions.
✗	Permanently remove the selected function(s).

Scene Editor

The scene editor, as its name suggests, is used to edit Scene functions. The editor is divided into tabs; with the first, **General** tab, you control the list of fixtures and channel groups that take part in the scene editing, together with the Scene name.
All subsequent tabs are used to control the individual channel values for each fixture and, if defined, the channels groups values.

General Tab Controls

On the left part of the screen, are displayed the buttons to control the fixtures used in the scene

Scene name	Change the name of the scene.
➕	Add an existing Fixture to the scene.
➖	Remove the selected Fixture(s) from the scene.
✓	Enable all channels of the selected fixtures.
🚫	Disable all channels of the selected fixtures.

On the right part of the screen, are displayed the buttons to control the channel groups used in the scene

✓	Enable all the selected channel groups.
🚫	Disable all the selected channel groups.

Hint: It's interesting to know that when a channels group is checked/unchecked, the fixtures controlled by the group will be automatically added to the left panel. Also, the channels controlled by the group will be automatically enabled/disabled on each fixture.

Channel Groups Tabs

This tab will be displayed only if one or more channel groups are selected in the General tab. Each Channel Group will be displayed with a quick macro access button (Click And Go if supported), a label with the group value, a fader, and the name of the group.

Fixture Tabs

Each fixture is represented by its own tab that contains sliders for each of the fixture's channels. Each channel can be enabled or disabled with a check box on top of the channel unit. The value of each channel can be set either by typing the value to the edit box on top of the slider or by moving the slider. Channels that provide multiple functions such as gobos, colors, etc. have also a button above the channel slider; this button can be used to directly select a specific function or

capability provided by that channel.
Hint: The keyboard shortcut to move between channels values edit boxes is '**Tab**'

Channel enabled/disabled status

If a channel has not been enabled, the scene will not touch that particular channel's value ever.

If a channel has been enabled, the scene will change that channel's value to a value defined in the scene. This is useful, for example, when you wish to control only the dimmer channel of a scanner fixture; you wouldn't want the scene to touch the scanner's pan, tilt, colour or gobo channels when you just want to fade in or fade out with the dimmer channel.

Controls

✓	Enable all channels from the current fixture.
🚫	Disable all channels from the current fixture.
⬅	Go to the previous tab in the view. If the first tab is selected, this will go to the last tab in the view. Keyboard shortcut: '**ALT+Left**'
➡	Go to the next tab in the view. If the last tab is selected, this will go to the first tab in the view. Keyboard shortcut: '**ALT+Right**'
⧉	The copy functionality has 2 modes: • Copy the values (and enabled/disabled states) of all channels in the current fixture to the clipboard. • Copy only the selected channels (CTRL-clicked, yellow background) values into the clipboard.
📋	Paste the values from clipboard to the current fixture. If no channels were selected during 'copy', then also the enabled/disabled states will be pasted
⧉	The 'copy all' functionality has 2 modes: • Copy all values (and enabled/disabled states) of all channels to all other fixtures taking part in the scene. • Copy only the selected channels (CTRL-clicked, yellow background) values to all the other fixtures taking part in the scene.
🎨	Launch a colour tool to select a specific colour and set that colour to the current fixture; This feature is enabled only for fixtures that are capable of CMY/RGB colour mixing.
🕒	Show/Hide the Speed Dial widget, used to facilitate the Scene parameters tuning like Fade In and Fade Out
⧉	Switch between "tab view" and "all channels view". The first mode will display one tab for each fixture, while the second will display a unique tab for all the fixtures
👁	Toggle blind mode for the selected fixture.

🔴	Clone the current scene and add a new step to the Chaser selected from the drop down list beside this button

Chaser Editor

The chaser editor, as its name suggests, is used to edit chaser functions.

Controls

Chaser name	Change the name of the chaser.
🕒	Show/Hide the Speed Dial widget, used to facilitate the Chaser parameters tuning like Fade In, Fade Out and Hold
▶	Start to run the Chaser to test the playback beginning from the selected step
■	Stop a previously started playback
⬅	When playback is active, go to the previous step in the Chaser
➡	When playback is active, go to the next step in the Chaser
⧉	Copy the selected steps into the clipboard.
📋	Paste the previously copied steps from clipboard to the current position selected. If none, the steps will be appended at the end of the list
✂	Remove the selected steps from the list and copy them into the clipboard.
➕	Add an existing Function to the chaser, using the Select Function dialog. The function are inserted after the currently selected step.
➖	Remove the selected step from the chaser.
⬆	Move the selected step above/before the previous step.
⬇	Move the selected step below/after the next step.
Run Order	• Loop: Run thru the steps over and over again. • Single Shot: Run thru the steps once and then stop. • Ping Pong: Run thru the steps over and over again, reversing direction at both ends.
Direction	• Forward: Run thru the steps from start to end; 1, 2, 3... • Backward: Run thru the steps from end to start; ...3, 2, 1
Fade In	• Common: Apply the same speed for all the steps in this Chaser • Per step: Apply a user defined speed for each step in this Chaser

Speed	• Per step: Apply a user defined speed for each step in this Chaser • Default: Apply the default speed for each step (0 seconds).
Fade Out Speed	• Common: Apply the same speed for all the steps in this Chaser • Per step: Apply a user defined speed for each step in this Chaser • Default: Apply the default speed for each step (0 seconds).
Step Duration	• Common: Apply the same duration for all the steps in this Chaser • Per step: Apply a user defined duration for each step in this Chaser

Show Editor

The Show Editor is a panel to display the current structure of a Show created with the Show Manager.
At the moment the Show Editor can only rename a Show, which is not possible in the Show Manager.

The tree view of this panel shows useful information regarding the displayed Show such as:

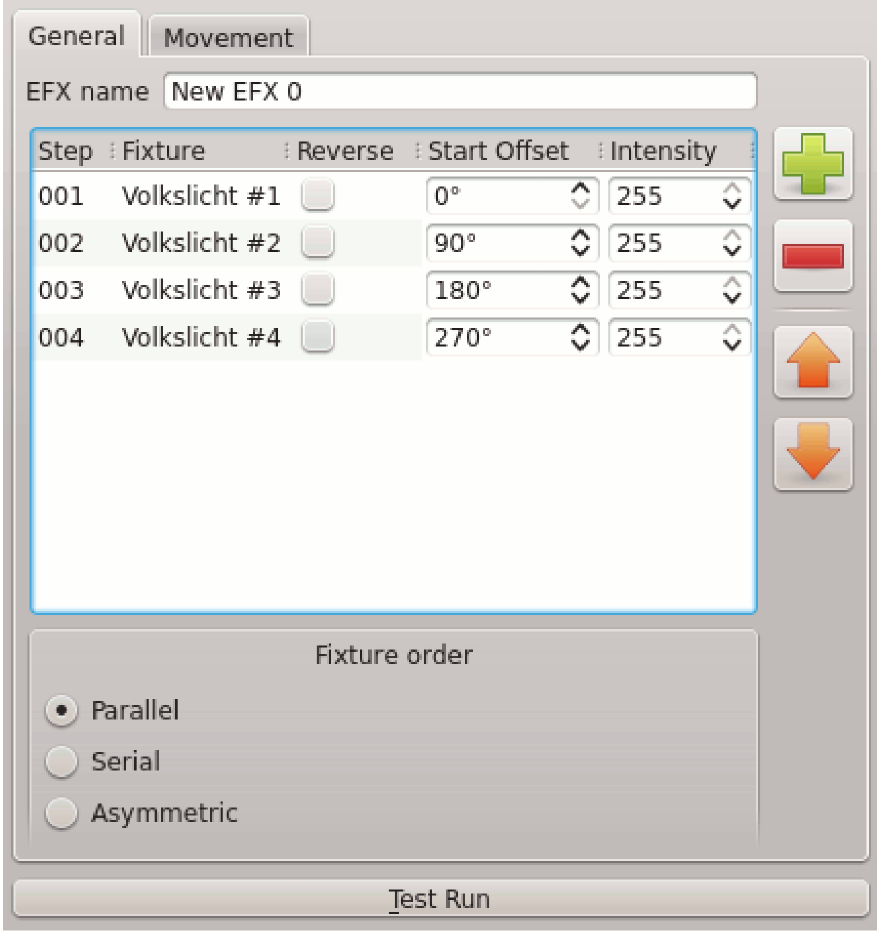
- **Function:** the function name
- **Steps:** the number of steps created in a sequence
- **Start Time:** the time a sequence is going to be played
- **Duration:** the duration of the sequence

EFX Editor

The EFX editor, as its name suggests, is used to edit EFX functions. The view is split into two tabs:

- **General** tab is for selecting Fixtures, speed and fixture order.
- **Movement** tab is for selecting details on how the fixtures should move their beams.

General Tab Controls



EFX name: New EFX 0

Step	Fixture	Reverse	Start Offset	Intensity
001	Volkslicht #1	☐	0°	255
002	Volkslicht #2	☐	90°	255
003	Volkslicht #3	☐	180°	255
004	Volkslicht #4	☐	270°	255

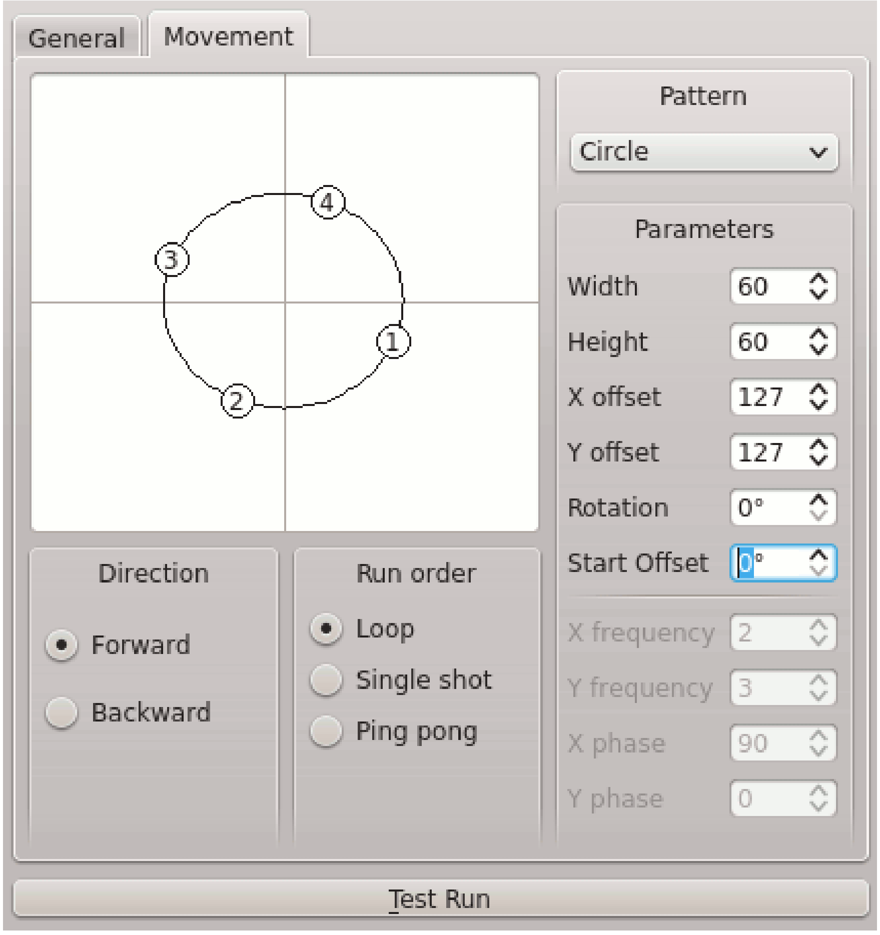
Fixture order

● Parallel
○ Serial
○ Asymmetric

EFX name	Change the name of the EFX.
Fixture list	• Step: Shows the order in which the fixtures start their movement in serial/asymmetric order • Fixture: The name of each fixture that has been added to the EFX • Reverse: When checked, the fixture reverses its movement as opposed to non-reversed fixtures. The "normal" direction is set by direction in the "Movement" page. • Start Offset: Value in degrees, where on the movement path this fixture should start • Intensity: The value set to the fixture's intensity/dimmer channel when the fixture starts to move
➕	Add a Fixture to the EFX, after the currently selected position. Since EFX is used for moving the head or mirror of an intelligent fixture, only fixtures with **pan and tilt** capability (8bit or 16bit) can be added to an EFX.

🟥	Remove the selected fixtures from the EFX.
⬆	Raise the selected fixture upwards (earlier) in serial order.
⬇	Lower the selected fixture downwards (later) in serial order.
Fixture order	Change the EFX's fixture order.
Speed	Change the EFX's speed settings.

Movement Tab Controls



Pattern: Circle

Width 60

Height 60

X offset 127

Y offset 127

Rotation 0°

Start Offset 0°

X frequency 2

Y frequency 3

X phase 90

Y phase 0

Direction: ● Forward ○ Backward

Run order: ● Loop ○ Single shot ○ Ping pong

Pattern area	Shows a 2D-projection of the fixtures' head/mirror movement. Every time you change a parameter, one small dot for each fixture travels the complete path in its selected direction from its start offset. The speed of the movement reflects selected speed settings.
Direction	Default direction of the fixtures (can be altered individually for each fixture by the checkbox in the **Reverse** column) • Forward: The fixtures move forwards along the pattern path • Backward: The fixtures move backwards along the pattern path
	• Loop: Run thru the steps over and over again.

Run order	• Single Shot: Run thru the steps once and then stop. • Ping Pong: Run thru the steps over and over again, reversing direction at both ends.
Pattern	Select the movement pattern algorithm.
Parameters	• Width: Choose the pan width (0-255) • Height: Choose the tilt height (0-255) • X Offset: Move the pattern's horizontal (pan) centerpoint (0-255) • Y Offset: Move the pattern's vertical (tilt) centerpoint (0-255) • Rotation: Rotate the pattern along its axis (0-360 degrees) • Start Offset: Where along the path the movement should start (0-360 degrees) • X Frequency: Change the **Lissajous** pattern's X (horizontal) frequency (0-5) • Y Frequency: Change the **Lissajous** pattern's Y (vertical) frequency (0-5) • X Phase: Change the **Lissajous** pattern's X (horizontal) phase (0-360 degrees) • Y Phase: Change the **Lissajous** pattern's Y (vertical) phase (0-360 degrees)

Note: when EFX is started, it will set the intensity of all included fixtures to the preset value. That usually means it set them full on. If you don't want this, just set the preset intensity to something lower, perhaps 0. That way you can control intensity of the fixture using other means like VC sliders for the intensity/dimmer channels, or by another chaser.

Collection Editor

The collection editor, as its name suggests, is used to edit collection functions.

Note: Collections don't have speed setting; each function you include in a collection follows its own speed settings.

Controls

Collection name	Change the name of the collection.
➕	Add an existing Function to the collection, using the Select Function dialog. The order of the functions has no practical meaning.
➖	Remove the selected functions from the collection.

RGB Matrix Editor

The RGB matrix editor, as its name suggests, is used to edit RGB matrix functions. The function works on predefined Fixture Groups created by user in Fixture Manager.

Controls

RGB matrix name	Change the name of the RGB matrix.
Fixture group	The Fixture Group that is controlled by this RGB matrix.
Preview area	Shows a preview of the currently selected pattern imposed on the fixtures defined in the currently selected fixture group. Note that the preview doesn't make difference between RGB-capable and monochrome/fixed color fixtures.
Pattern	Select the pattern and color that are used on the selected fixture group for drawing graphics.
Animated Text	Available only when **Text** is the selected pattern. • The text edit field is used to edit the text content that is scrolled/flashed on the matrix. • The font button Aa is used to select the font (tip: bitmap fonts work best) • The drop-down box is used to select the animation style (Horizontal, Vertical or Letters)
Offset	Available only when **Text** is the selected pattern. • X: Used to shift the pattern along the horizontal x-axis (negative values go left, positive values go right) • Y: Used to shift the pattern along the vertical y-axis (negative values go up, positive values go down)
Run Order	• Loop: Run thru the steps over and over again. • Single Shot: Run thru the steps once and then stop. • Ping Pong: Run thru the steps over and over again, reversing direction at both ends.
Direction	• Forward: Run thru the steps from start to end; 1, 2, 3... • Backward: Run thru the steps from end to start; ...3, 2, 1
Speed	Change the RGB matrix' speed settings.
Test Run	Make the RGB Matrix run as if started from a Virtual Console button. Note that the preview stops while the function is running.

RGB Script Fundamentals

Users can write their own RGB scripts to produce custom graphics projection with the RGB Matrix function. The scripts' operation principle is to produce a number of RGB maps that each represent one step in the graphics animation. The principle is basically the same as in movies: the audience observes a moving picture, which in reality is only a sequentially-played stream of static images that only produce an illusion of movement.

The scripts themselves are written in ECMAScript, which is also known as JavaScript. Note that the language is case-sensitive and as a de-facto standard follows camel case rules, so make sure you write everything correctly and pay special attention to the required API features.

Script files should be named after the script's name and must have a **.js** extension. Depending on platform, the files should be placed either in the QLC+ system script directory or, preferably, the user script directory:

- **Linux user dir:** ~/.qlcplus/rgbscripts/
- **Linux system dir:** /usr/share/qlcplus/rgbscripts/

- **OSX user dir:** ~/Library/Application Support/QLC+/RGBScripts
- **OSX system dir:** /Applications/QLC+.app/Contents/Resources/RGBScripts

- **Windows user dir:** Not applicable
- **Windows system dir:** C:\QLC+\RGBScripts

RGB Script API

Foundation

The scripts must be self-executing, i.e. when they are evaluated, the script itself is put inside an anonymous function that executes itself and returns an object that contains the required API functions:

```
(
function() { // Anonymous function starts here
    var algo = new Object;
    return algo; // Return the script object
} // Anonymous function ends here
)() // Anonymous function is executed here upon evaluation
```

Properties

However, a script with nothing more than an empty object does nothing, no matter how self-executing it might be. You must also declare some **properties** for the returned object so that QLC+ knows how to use the script and to show it to the user (you). So, you need to declare the following properties for the returned script object:

- **apiVersion:** The API version that the script follows. Currently '1' is correct number. 0 will cause the script to be treated as invalid.
- **name:** The name of your script. This name appears in the pattern selection box in the RGB Matrix Editor
- **author:** The name of the person who has written the script. **You.**

With this in mind we add declarations for these three properties to the script:

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 1; // Must be currently '1'
    algo.name = "My cool RGB script";
    algo.author = "Your name";
    return algo;
}
)()
```

Functions

Now we are getting to the actual business of producing data for the RGB Matrix. The current API version uses two functions to achieve this:

- rgbMapStepCount(width, height)
- rgbMap(width, height, rgb, step)

No assumptions on the calling order or argument immutability should be made, i.e. do not cache the values from either function and assume that they are valid until the worlds end. The parameters might change at any point (usually when the user changes the matrix size, color or direction) thus invalidating any cached values.

rgbMapStepCount(width, height)

When QLC+ calls this function, it wants to know the number of different RGB maps returned by the rgbMap() function, when the RGB matrix size is **width** times **height** pixels. It must always return the same result when called with the same **width** and **height** parameters again and the result must not change over time.

Parameters:

- **width:** The width of the grid
- **height:** The height of the grid
- **Return value:** Number of steps produced by rgbMap() with the given width and height parameters

So, now we add this support function to the script:

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "My cool RGB script";
    algo.author = "Your name";

    algo.rgbMapStepCount = function(width, height) {
        ...
        return number_of_steps_when_width_is_oranges_and_height_is_jabberwocky;
    }

    return algo;
}
)()
```

rgbMap(width, height, rgb, step)

This function is the actual brains of the script. It produces two-dimensional arrays whose size MUST be **height** times **width**. I.e. the array returned from this function must contain **height** items and each of these items must be an array that contains **width** items that must be 32bit integers, representing an RGB color as understood by QRgb without alpha channel (0x00RRGGBB). The **rgb** parameter is an integer-representation of the color selected by user in the RGB Matrix Editor. The **step** parameter tells the step number requested by the RGB Matrix function and is guaranteed to be within (0, rgbMapStepCount(w, h) - 1).

Parameters:

- **width:** The width of the grid
- **height:** The height of the grid
- **rgb:** The color selected by user
- **step:** Current step number to produce (between 0 and rgbMapStepCount(w, h) - 1)
- **Return value:** An array containing [height] arrays each containing [width] integers

Just like the previous function, we also add this other one to the script. Now we have a full and ready template for any RGB script for your indulgence.

```
(
function() {
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "My cool RGB script";
    algo.author = "Your name";

    algo.rgbMapStepCount = function(width, height) {
        ...
        return number_of_steps_when_width_is_oranges_and_height_is_jabberwock;
    }

    algo.rgbMap = function(width, height, rgb, step) {
        ...
        return a_2d_array_of_arrays;
    }

    return algo;
}
)()
```

Development Tool

There is a development tool available in the QLC+ source repository that makes it easier to debug & test your custom scripts with a web browser. To use the tool, you must download the following two files to a directory on your hard disk, open the **devtool.html** file with your browser and follow its instructions:

- devtool.html
- devtool.js

(Right-click and "Copy Link Location" works probably best)

Example Script: Full Columns

```
/*
  Q Light Controller Plus
  fullcolumns.js

  Copyright (c) Heikki Junnila

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0.txt

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
*/

(

/**
 * This algorithm produces fully-lit columns, meaning all pixels on a single
 * column are lit together.
 */
function()
{
    var algo = new Object;
    algo.apiVersion = 1;
    algo.name = "Full Columns";
    algo.author = "Heikki Junnila";

    /**
     * The actual "algorithm" for this RGB script. Produces a map of
     * size($width, $height) each time it is called.
     *
     * @param step The step number that is requested (0 to (algo.rgbMapStepCount - 1))
     * @param rgb Tells the color requested by user in the UI.
     * @return A two-dimensional array[height][width].
     */
    algo.rgbMap = function(width, height, rgb, step)
    {
        var map = new Array(height);
        for (var y = 0; y < height; y++)
        {
            map[y] = new Array();
            for (var x = 0; x < width; x++)
            {
                if (x == step)
                    map[y][x] = rgb;
                else
                    map[y][x] = 0;
            }
        }

        return map;
    }

    /**
     * Tells RGB Matrix how many steps this algorithm produces with size($width, $height)
     *
     * @param width The width of the map
     * @param height The height of the map
     * @return Number of steps required for a map of size($width, $height)
     */
    algo.rgbMapStepCount = function(width, height)
    {
        // Each column is lit completely at a time, so because there are $width
        // columns in the map, the number of steps must be $width to light all
        // columns per round.
        return width;
    }

    return algo;
}
)()
```

Function Wizard

The Function Wizard's purpose is to help users create some common functions for a quick start.

Controls

Fixture List	List of fixtures that will be included in the functions created with the wizard. You can see each fixture's capabilities for Colour-, Gobo- and Shutter- type scenes on the **Supported capabilities** column.
➕	Include Fixtures to the functions that will be created with the wizard.
➖	Remove the selected fixtures from the function to be created.
Scenes	• Colours: Create scenes for changing the colour simultaneously to all currently included fixtures. For this to work, the fixtures must have at least one colour channel that controls a fixed-color wheel. • Gobos: Create scenes for changing the gobo simultaneously to all currently included fixtures. For this to work, the fixtures must have at least one gobo channel. • Shutter: Create scenes for changing the shutter position simultaneously to all currently included fixtures. For this to work, the fixtures must have at least one shutter channel.

Select Function(s)

Whenever a Function needs to be selected, for example when you're adding steps to a Chaser, using the Chaser Editor, the **Select Function** dialog is used.

Sometimes the place you are selecting the function to, might allow selecting multiple functions, for example, when adding steps to a Chaser or a Collection. In those cases, the order in which the functions are selected will also be the order in which they are added to their destination. Oppositely, when attaching a function to a button in the Virtual Console, you may select only one function at a time.

Refer to your operating system manual for documentation on how to select multiple items in a list.

Controls

Functions to display	There are two radio buttons on the top of the window: **All functions** and **Running functions**. By clicking on them, the list below will be filled only with the desired functions. For example, when performing a Live edit, you might want to display only the running functions. Obviously, in design mode, selecting "Running functions" will show an empty list.
Function list	• Name: The names of available functions • Type: The type of each of the available functions
Filter	You can filter the function list so that only those function **types** that have been checked are shown. Filtering is especially useful if you have lots of functions to select from. • Scenes: Display scene functions in the selection list • Chasers: Display chaser functions in the selection list • EFX's: Display EFX functions in the selection list • Collections: Display collection functions in the selection list • RGB Matrixes: Display RGB Matrix functions in the selection list • Show: Display Show functions in the selection list

Select Fixture(s)

Whenever a Fixture needs to be selected, for example when adding fixtures to a Scene function, the **Select Fixture(s)** dialog is used.

The dialog is very straightforward; there is only a list of fixtures that you can select from. Sometimes the place you are selecting the fixture to, might allow selecting multiple fixtures, for example, when adding fixtures to a Scene or an EFX. In those cases, the order in which the fixtures are selected will also be the order in which they are added to their destination.

Refer to your operating system manual for documentation on how to select multiple items in a list.

🎬 Show Manager

The Show Manager has been introduced in QLC+ starting from version 4.0.0, after forking the original QLC code on November 5th, 2012.

This feature is meant to give users the possibility to setup a time driven show in a friendly and completely graphical way.
The graphic interface shows a multitrack view, typical of audio sequencers or video editing softwares, and with it users are allowed to place QLC+ Functions at the desired place and time in the view.

A typical use case are those gigs where bands play songs following a metronome and the light show has always to be the same, following the music.
Another case are visual entertainment shows, where dancers or singers follow some music and lights create the atmosphere at the right time.

Controls

🎬	Create a new Show. A Show is represented as a multitrack workspace where tracks, sequences and audio elements can be added
Shows list	This drop down box list the currently created Shows. Clicking on a Show will display it
🟥🟩 🟦🟨	Create a new track. In QLC+ a track corresponds to a Scene. A Show can virtually have an infinite number of tracks.
●●●	Create a new Sequence item. A sequence is a special Chaser bound to a Scene. All the steps created in a sequence will only act on the selected track's DMX channels
♫	Create a new audio item. An audio item simply represents an audio file. Audio items can be added on any track, but if you intend to create a separate track for it, just create an empty Scene track. **Note:** It is possible to display the waveform preview of an audio item just by right clicking it and selecting the desired channels you want to display (Mono, left channel, right channel, stereo) **Warning:** Even though QLC+ allows you to, it is not possible to play two audio files simultaneously. Especially on Windows, you might experience unwanted crashes.
❐	Copy the currently selected item into QLC+ clipboard
📋	Paste QLC+ clipboard content at the cursor position. When performing this operation two checks are performed: • Overlapping: checks if the item you're going to paste overlaps with an existing item in the selected track • Validity: In case you're pasting a Sequence, QLC+ will verify if the Sequence contents are compatible with the currently selected track

❌	Delete the currently selected item. This can be a sequence, an audio item or a track. Note that deleting a track will delete also all its sequences/audio children. **Note: Show Manager will only perform a "visual removal" of functions. To permanently delete them, please use the Function Manager**
🎨	Assign a custom color to the selected item
▦	Enable/disable the "Snap to grid" functionality. The view will be filled of vertical bars corresponding to the header markers (time or BPM) Snapping to grid will correct your items dragging to the nearest reference bar
00:00:00.000	This field displays the cursor time position either if the playback is stopped or activated
▶	Start the current Show playback from the cursor position
■	Stop the current Show playback. Clicking once will stop the cursor at the current position for later resume. Clicking twice will restore the cursor to 0 seconds. **Hint:** When resuming a show with audio tracks, please keep in mind that audio accuracy depends on the file formats you are using. For example MP3 file resume is not accurate, while WAV is.
Time markers	This drop down menu lets you to choose the desired time division to display for your Show. This can be either 'Time'. 'BPM 4/4', 'BPM 3/4' or 'BPM 2/2'
BPMs	When a BPM time division mode is selected, this field lets you decide the appropriate BPM to set for you Show. This ranges from 20 to 240. This can be quite useful when dealing with electronic music or BPM synchronized shows

Just 4 steps

The Show Manager has been studied to be used quickly and easily. Basically with just 4 steps a complete Show can be created:

1. Add a new Show

First of all you need to add a new Show to the view. This creates an empty multitrack view with no tracks and no items, ready to be filled.
A popup will appear asking the name to assign to the Show. It is possible to change the name afterward with the Function Manager, using the Show Editor panel.

2. Add a track

In the Show Manager, a track corresponds to a Scene. When you add a track, a popup will ask you to select an existing Scene or to create a new one with a default name.
When done, a new track will be created. All the sequences created on this track will act only on the associated Scene, not affecting all the other tracks.

A newly created track will be set automatically as active. An active track has a green light on the left side.
A Track can be set to mute and solo states. Mute state will exclude the track from the playback, while the solo state will mute all the other tracks of the Show.

When right clicking on a track, it is possible to move it up or down for logical ordering.
Once selected, a track will display its Scene Editor on the bottom of the screen.

3. Add a Sequence or Audio

When a track has been activated, it is possible to add a Sequence or and Audio item to it.
A new item will be placed at the cursor position, but it is always possible to move it to the desired time just by dragging it along the timeline.
A show item cannot be moved to another track, since it is bound to the track where it has originally been created.

A show item can be copied , pasted or deleted with the toolbar icons. The pasted item will be placed at the current cursor position.

The background color can be changed with the toolbar icon . The assigned color will be saved in your project file.
Once selected, a sequence will display its Chaser Editor on the right side of the screen.
When right clicking an item a contextual menu will be displayd, showing the following option:

- Preview: This option is available only on audio items. It will display the waveform preview for right, left or stereo channels when available
- Align to cursor: This option will move the selected item to the cursor position

Follow step 4 to understand how to fill a Sequence

4. Add some steps

Once a sequence has been created, it is now the time to add steps to it.
Please note that the difference between a Chaser and a Sequence is that a Sequence is bound to the Track (so the Scene) where it has been created. So, when adding a step, QLC+ will not ask the Function to associate, but will always use the channels of the same Scene.
When modifying a step duration, fade in or fade out time, you will see the sequence changing on the multitrack view, giving an immediate feedback for aligning to other sequences or certain points in time.
Fade In and Fade Out times will be displayed as diagonal lines over the sequence item on the multitrack view, while different steps will be divided by vertical lines.

To increase the complexity of the Show, more sequences, audio and scenes can be added. Just repeat the above steps depending on your needs.

And finally...play !

When a complete show has finally been created, it can be played just by clicking on the Play icon.
At the moment playback starts always from the beginning of a Show and the Stop button will bring the cursor back to 0.

Virtual Console

The purpose of the Virtual Console is to act as an open sandbox where the user can create a lighting desk layout of his choice. Users can place various GUI (Graphical User Interface) elements, called widgets, onto the console surface:

- Buttons for starting and stopping functions
- Sliders for adjusting channel values or function intensity
- Speed dials for adjusting function speeds
- XY Pads for manually moving intelligent light beams
- Cue lists for theatrical performance purposes
- Frames for grouping various widgets together
- Solo Frames for keeping only one button/function active at a time
- Labels to act as static information banners for other widgets

The topmost part of the Virtual Console is dedicated to a tool bar that provides quick access to the most common actions; adding new widgets, copying them to/from the clipboard, configuring the widgets and, in case of emergencies, a panic button for stopping all currently running functions.

Each Virtual Console widget's style can be configured (to some extent), widgets can be moved anywhere and resized to any size on the Virtual Console canvas. See the article on Virtual Console Widget Styling & Placement.

Tool Bar Controls

	Add a new Button to the currently selected frame.
	Add a new Button Matrix to the currently selected frame, using the Add Button Matrix dialog.
	Add a new Slider to the currently selected frame.
	Add a new Slider Matrix to the currently selected frame.
	Add a new Knob to the currently selected frame. This is a convenience shortcut to add a slider widget using the knob style.
	Add a new Speed Dial to the currently selected frame.
	Add a new XY Pad to the currently selected frame.
	Add a new Cue List to the currently selected frame.
	Add a new Frame to the currently selected frame.
	Add a new Solo Frame to the currently selected frame.

	Add a new Label to the currently selected frame.
🎤	Add a new Audio triggers widget to the currently selected frame.
10 27	Add a new Clock widget to the currently selected frame.
✂	Cut the currently selected widget(s) to the clipboard.
⧉	Copy the currently selected widget(s) to the clipboard.
📋	Paste the widget(s) in clipboard to the currently selected frame.
✗	Destroy the currently selected widget(s) completely. If a frame has been selected, this destroys everything inside it, including other frames.
✏	Configure the currently selected widget with its own configuration dialog.
I	Rename the currently selected widget(s).
⬆	Bring the selected widget to front.
⬇	Send the selected widget to back.
🎨	Change the background color of the selected widget.
🖼	Set a background picture for the selected widget.
abc	Change the font color of the selected widget.
Aa	Set the font properties of the selected widget.
🛠	Configure the Virtual Console properties like the size workspace, the widgets default properties and Grand Master
⊗	Stop all currently running functions.

Virtual Console Button

A Button is the simplest and at the same time the most powerful widget in QLC+; with it, you can start, stop and flash your functions.

Configuration

Buttons can be configured with the properties ✎ button found in the toolbar or by double clicking the button itself.

Button label	Set the friendly name of the button. The name appears on the button in Virtual Console. **NOTE:** If you set an icon to a button with the **Edit -> Icon -> Choose** menu, the icon overrides this label. second column title.
Function	Shows you the name of the Function that has been currently assigned to the button. Attach a function to the button Detach the current function from the button
Attributes	Shows you the list of attributes that the assigned function exposes. Attributes can be controlled by right clicking on the button when in Operate Mode. A popup will display a number of sliders corresponding to the number of available attributes. In case the assigned function is a Show, attributes correspond to the Show tracks, so you can tune the intensity (or volume) of a whole track with just one click.
External input	You can attach an external input channel from an input device (like a slider board) to buttons so that you don't always have to use the mouse, touch or keyboard to access the buttons. • Input universe: The input universe that you wish to provide input data to the button. • Input channel: The individual input channel within the selected input universe that you wish to use for controlling the button. • Auto Detect: When toggled, you can just press a button on your external input hardware and it will be automatically assigned to the button. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first. • Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.
Key combination	You can attach a keyboard key (combination) to the button, which then acts as if you clicked the button directly with your mouse. Attach a keyboard key to the button Detach the current key combination from the button

On button press...	- Toggle function on/off: When you click the button, the attached function is started. When you click the button a second time, the function is stopped, unless it has already stopped by itself. - Flash function: You can "flash" the attached Scene when you keep the button pressed. If another type of function is attached to the button, nothing happens when you click it. - Blackout on/off: When you click the button, QLC+ will toggle the blackout mode - Stop all functions: When you click the button, all the QLC+ running functions will be stopped immediately
Adjust function intensity	If checked, this feature will adjust the assigned function intensity right before playing it when the button is pressed.

Virtual Console Button Matrix

A Button Matrix is basically just a way for quickly creating multiple Buttons at a time inside a common container Frame.

When creating a new button matrix, you can assign functions to each of the buttons quite quickly with the Add Button Matrix dialog.

Configuration

Refer to Frame and Button on how to configure them.

Virtual Console Slider

Sliders are used for two distinct purposes: Fixture channel level setting and Function playback & intensity adjustment. Any slider can operate on either of these modes and each mode has its own configuration options.

Configuration

Sliders can be configured with the properties button found in the toolbar or by double clicking the slider itself.

Configuration - General Tab

The General tab holds all of the sliders properties that are shared between the slider's two modes.

Slider name	Change the slider's name.
Widget appearance	A slider can be displayed as a vertical fader or as a knob . Appearance can be changed on the fly.
Value display style	• Actual: Display actual (DMX) values • Percentage: Display percentage values (0-100%)
Slider movement	• Normal: Values increase towards the top and decrease towards the bottom. • Inverted: Flip the slider upside down so that values increase towards the bottom and decrease towards the top.
External input	You can attach an external input channel from an input device (like a slider board) to sliders so that you don't always have to use the mouse or touch to move sliders. • Input universe: The input universe that you wish to get input data to the slider. • Input channel: The individual input channel within the selected input universe that you wish to use for controlling the slider. • Auto Detect: When toggled, you can just move/press a button/slider/knob on your external input hardware and it will be automatically assigned to the slider. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first. • Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.

Configuration - Level Tab

If the slider is not currently in Level mode, all you see is a button telling you to click it to switch the slider to Level mode. Clicking it will also show the actual Level mode properties.

Value Range	• Low limit: Set the lowest DMX value that the slider can set • High limit: Set the highest DMX value that the slider can set • From capability: You can limit the slider's level value range to a certain capability range within a fixture's channel. When you have any channel value range selected from the fixture list (for example *"Dimmer Control 006 - 128"*) and you click this button, the **low and high limit** of this slider are taken automatically from that capability and are set to 6 and 128, respectively.
Fixture list	You can select individual channels from this list that contains all channels from all of your fixtures. Those channels that you have placed the tick mark on, will be controlled by this slider.
All	Clicking this button will select ALL channels from ALL fixtures.
None	Clicking this button will clear ALL channel selections concerning ALL fixtures.
Invert	Invert the current selection. If you have channels 1, 3 and 5 selected from all fixtures, clicking this button will de-select channels 1, 3 and 5 and instead select channels 2, 4 and 6 from those fixtures.
By group...	If you wish to control a specific channel group from ALL fixtures, you can click this button and select the channel group to control. Accepting the dialog will select ALL channels that belong to the selected group from ALL fixtures. This is particularly useful if you wish to have common control over, for example, all of your scanners' *intensity* channels.
Click & Go	Click & Go is a QLC+ technology which allows you to quick access fixtures functionalities in a completely visual way. When Click & Go is enabled, a button will appear at the bottom of the slider and with just 2 clicks you will reach the desired result. The supported modes are: • None: Click & Go disabled. No button will be displayed. • Color: Single color selection. A gradient of the color controlled by the slider will be displayed, allowing you to jump to the desired color visually. • RGB: RGB color selection. A RGB Color picker will be displayed, allowing you to pick any color from black to white. 16 preset colors are displayed on the left side for convenience. When selecting a color, the slider value will be places half way (128). Moving it downward will fade to black, while moving it upward will tend to white. • Gobo/Effect: Gobo/Macro selection. A grid view of the fixture defined macros will be displayed and you will be able to choose a macro from its color, its gobo or its name. When the mouse is over a macro cell, a blue bar will appear, allowing you to choose also the intermediate value within the same macro. This is useful for functionalities like Gobo rotation speed.

Configuration - Playback Tab

If the slider is not currently in Playback mode, all you see is a button telling you to click it to switch to Playback mode. Clicking it will also show the actual Playback mode properties.

When the slider is in playback mode, the slider acts like a combined button and a slider; you start a function AND simultaneously control the function's intensity with the slider. When the slider is a zero, the function is stopped and any value above zero will start the function (unless already started) and simultaneously adjusts the function's intensity (if applicable).

Function	Displays the function that is currently attached to the slider.
🔌✓	Attach a function to the slider.
🔌	Detach the currently-attached function.

Virtual Console Slider Matrix

A Slider Matrix is basically just a way for quickly creating multiple Sliders at a time inside a common container Frame.

Configuration

Refer to Frame and Slider on how to configure them.

Virtual Console Speed Dial

Speed dials are meant for adjusting the speed of a set of functions from virtual console, much like bus-mode sliders in the previous versions of QLC+. While the bus concept relied on assigning functions and a slider to the same bus, the speed dial's method is a simpler and more straightforward one: User directly selects a set of functions whose speed properties are controlled with the dial, thus removing the need for buses in the middle.

Operation

The speed dial itself looks and works just the same as the speed dials in various function editors: There is a **big dial** in the middle that increases the time when rotated clockwise and respectively, decreases it when rotated counter-clockwise. The plus ➕ and minus ➖ buttons perform the same thing.

The **current time** value is displayed in four parts. When any of these fields is clicked, the dial and the plus/minus buttons will increase/decrease that part of the time. The mouse wheel can also be used in place of the dial/buttons and the fields accept also values written on the numeric keypad.

- **h**: Hours
- **m**: Minutes
- **s**: Seconds
- **.xx**: Centiseconds (100ths of a second)

Clicking the **infinite** button disables all of the dial's controls and sets the currently adjusted functions' speed to infinity. For example, to a Chaser, an infinite duration means that it will never switch to the next step unless manually instructed to (see Cue List).

The **tap** button can be used to adjust the time to a beat. Practically the dial measures the elapsed time between clicks and sets this time to the controlled functions as if the dial was adjusted accordingly.

For cuelists/chasers and RGBMatrixes, the tap button advances to next scene/position when speed dial is set to adjust duration, and the tapped tempo is more than 1/4 of the original duration.

Configuration

Speed dials can be configured with the properties ✎ button found in the toolbar or by double clicking the speed dial itself.

Speed Dial Name	Change the dial's name.
Function list	Displays the list of functions, whose speed value(s) are being controlled by the dial.
➕	Add function(s) to be controlled by the speed dial.
➖	Remove the selected functions from the speed dial's list of controlled functions.

Speeds to Adjust	• **Fade In:** Adjust the functions' fade in speed • **Fade Out:** Adjust the functions' fade out speed • **Duration:** Adjust the functions' duration

Virtual Console XY Pad

An XY Pad is for manually controlling the **pan** and **tilt** channels of intelligent lighting fixtures, namely scanners and moving heads. The pad is a resizable canvas with a handle and two sliders that you can grab to obtain direct access to the attached fixtures' mirror/head position.
Starting from QLC+ version 4.5.0, two more sliders have been introduced to give you the possiblity to set a working range window where your moving heads will operate.
Note that from the GUI the handles will be bound into the green window area, while with an external controller all the values will be scaled to the window, so you will be able to have more sensitivity to control your fixtures.

Configuration

XY Pads can be configured with the properties ✏ button found in the toolbar or by double clicking on the XY pad itself.

XY Pad name	Set the name of the XY Pad. The name is shown in the upper-left corner of the widget on Virtual Console.
Y-Axis slider movement	The appearance of the vertical slider (namely Y-Axis control) can be set to **Normal** or **Inverted**. In the first case the maximum value will be reached on the bottom of the pad, while in the second case it will be reached on the top.
Fixture tab	Shows the fixtures that are currently controlled with the XY Pad. • Fixture: The names of each fixture • X-Axis: Shows the value range (and reversal if applicable) of the horizontal (pan) axis for each fixture • Y-Axis: Shows the value range (and reversal if applicable) of the vertical (tilt) axis for each fixture
➕	Add Fixtures to be controlled by the XY Pad, using the Select Fixture dialog. **NOTE:** only such fixtures that have pan and tilt capabilities are shown in the dialog.
➖	Remove the selected fixtures from the XY Pad's control list.
✏	Edit the selected fixture's behaviour with the XY Pad.

Fixture movement configuration

You can change the extent of movement and reverse fixtures' movement with the XY Pad Fixture Configuration dialog.

Horizontal X-Axis	• Minimum: Set the minimum pan limit for the fixture • Maximum: Set the maximum pan limit for the fixture • Reverse: Reverse fixture's movement on the horizontal (pan) axis

Vertical Y-Axis	• Minimum: Set the minimum tilt limit for the fixture • Maximum: Set the maximum tilt limit for the fixture • Reverse: Reverse fixture's movement on the vertical (tilt) axis

External input configuration (Input Tab)

You can attach an external input channel from an input device (like a tablet) to XY pads so that you don't always have to use the mouse to control your fixtures.

- Input universe: The input universe that you wish to get input data to the XY pad.
- Input channel: The individual input channel within the selected input universe that you wish to use for controlling the XY pad.
- Auto Detect: When toggled, you can just move/press a button/slider/knob on your external input hardware and it will be automatically assigned to the XY pad. The latest combination is shown on the text boxes when QLC+ receives input data. If you don't see anything in the boxes, your input connection might have a problem that you need to fix first.
  Hint: When assigning a XY pad from OSC, you need to click the "Tilt / Vertical axis" auto detect button, otherwise the X/Y axis will result inverted.
- Choose...: Shows the Select Input Channel dialog that you can use to select an input channel manually.

Virtual Console Cue List

A Cue List provides a list of Functions that you can step through with a single keyboard key. The Cue List is at home in theatrical shows where the lighting operator needs only to follow the script and toggle the next cue as the performance goes on.

The first column on the Cue List displays the cue number that runs from 1 to infinity and beyond . The second column shows the individual Function name that has been assigned to that particular cue.

Please note that only Chasers can be assigned to a Cue List, for the simple reason that the Chaser is the only function with a duration. Since any other Function can be added as a step to a Chaser with the Chaser Editor, it is simple to create the desired Cues by mixing Scenes, Collections, and so on...

Configuration

Cue Lists can be configured with the object properties button ✎ found in the toolbar or by double clicking on the Cue List widget.

Cue list name	Set the friendly name of the Cue List. The name appears as the Cue List's second column title.
Cue list	This panel allows the user to select the Chaser to associate to the Cue List.
➕	Associate a Chaser to the Cue List using the Select Function dialog.
➖	Remove the selected Chaser from the Cue List.
Next cue key combination	You can select the key that you want to use to skip to the next cue in the Cue List with this option. The key combination is shown on the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Previous cue key combination	You can select the key that you want to use to skip to the previous cue in the Cue List with this option. The key combination is shown on the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Stop cue key combination	You can select the key that you want to use to stop the cue in the Cue List with this option. The key combination is shown on the text field; if the field is empty, there is no key combination currently attached to the Cue List.
Crossfade key combinations	You can select the keys that want you use to control the Cue List crossfade functionality. The key combinations are shown on the text field; if the field is empty, there are no keys combination currently attached to the Cue List.
⌨	Assign a keyboard key to the specific action to be performed on the Cue List.
✖	Clear the key binding.

Operate mode

When switching QLC+ to Operate Mode, the Cue List will become active, allowing the selection of the desired steps included in the associated Chaser.

Four buttons are displayed at the bottom of the Cue List widget:

▶ ■	Start/stop the selected action in the Cue List
⬅	The previous action in the Cue List will be activated. When reaching the top, the previous action will be the last of the Cue List
➡	The next action in the Cue List will be activated. When reaching the bottom, the next action will be the first of the Cue List
🎚	When clicked, this button will open/close a side panel showing two sliders. Each slider is connected to an action and to understand their meaning we'll call them "current action" and "next action" sliders. The label on the bottom will show the corresponding color and number of the running action. The checkbox on the top will link the sliders together for a linear transition When the Cue list is running the sliders will control the intensity of the associated actions, overriding their Fade In speed, to allow manual control of the transition between two actions. When the current action slider will reach zero value, the next action slider will turn to blue, to indicate the action has been stopped and the Cue list moved to the next action. At this point the next action slider becomes the current action and what was current action becomes next action. In this way it is possible to manually crossfade two sequential actions.

Hint

When a step is added to a Chaser using the Chaser Editor, the default duration is set to 0.
To avoid your actions in the Cue List to loop endlessly without any result, please set the duration of your steps, either by double clicking on the duration field or by using the Speed Dial widget 🕒
Please note that if you need light scenes that you're going to change manually with the Cue List widget, you might want to set the duration to "infinite" (∞) using the Speed Dial widget enabled in the Chaser Editor by clicking on the 🕒 button.

Virtual Console Frame

A Frame is a container that can hold other widgets and other frames inside. In fact, the whole Virtual Console is already your bottom-most frame. Also, if you apply some styling attributes to a frame, all of its children (that don't have their own non-default style settings) also inherit their parent's style properties.

Configuration

Frames can be configured with the properties ✎ button found in the toolbar or by double clicking the frame itself.

Other than the standard styling & placement options, Frames have the following additional options, divided into 2 tabs:

Appearance tab

- **Frame name**: Allow you to assign an arbitrary label to the Frame. This will be displayed only when the "Show header" option is enabled (see below)
- **Accept child widgets**: Allow you to add widgets to the Frame
- **Allow resizing**: Allow the Frame geometry to be resized
- **Show header**: Display a useful header composed by a button and a label. The button allows you to expand/collapse the Frame, to save a lot of Virtual Console space. On the label the Frame name will be displayed

Pages tab

Starting from version 4.5.0, QLC+ gives the possibility to turn Virtual Console Frames into multi-page widgets, useful when dealing with a lot of widgets or controllers supporting pages.
The multi-page functionality is disabled by default.
Important note: this functionality will work only with an input profile with a previous page ⬅ and next page ➡ buttons set
Following the options to configure this functionality:

- **Enable**: This check box enable/disable the multi-page functionality. When enabled, VC frames will display additional controls in the header bar (if enabled). These are: Previous page button, page number label, next page button.
- **Number of pages**: With this counter it is possible to determine the number of pages the frame will handle.
- **Clone first page widgets**: When checked, QLC+ will clone the widgets of the frame's first page into all the other new pages defined by the "Number of pages" field. This is a very useful option to speed up an external controller mapping, where all the pages have the same widgets.
- **External input - Previous page**: Here it is possible to set an external input signal or a keyboard combination to display the previous page of the frame.
- **External input - Next page**: Here it is possible to set an external input signal or a keyboard combination to display the next page of the frame.

Virtual Console Solo Frame

A Solo Frame is almost exactly the same kind of a container as a normal Frame that can hold other widgets and other frames inside. However, the difference with Solo Frame is that it treats any Buttons inside it a bit differently by allowing only one button to be enabled at a time. For example, consider you have **Button A** and **Button B** inside a Solo Frame with **Button A** currently enabled. Next, you click button **Button B** which automatically results in **Button A** being released, leaving now only **Button B** enabled.

Configuration

Solo Frames can be configured with the properties button found in the toolbar or by double clicking the solo frame itself.

Other than the standard styling & placement options, Solo Frames have the following additional options:

- **Frame name**: Allow you to assign an arbitrary label to the Solo Frame. This will be displayed only when the "Show header" option is enabled (see below)
- **Accept child widgets**: Allow you to add widgets to the Solo Frame
- **Allow resizing**: Allow the Solo Frame geometry to be resized
- **Show header**: Display a useful header composed by a button and a label. The button allows you to expand/collapse the Solo Frame, to save a lot of Virtual Console space. On the label the Solo Frame name will be displayed

Virtual Console Label

A Label is just a simple text field that can be placed for additional information, for example as a heading for a group of buttons or the title of a frame.

Configuration

You can only set the displayed text content for a Label with the **Rename** button. It does not have any other purpose nor configurable options, other than the standard styling & placement options that all Virtual Console widgets have.

Virtual Console Audio Triggers

Starting from QLC+ version 4.4.0, this functionality allows you to use an audio input source like a microphone, to add more life to your light shows.

1 Introduction

When clicking on the 🎤 icon, an audio trigger widget will be added to your Virtual Console.
The widget graphics area shows the live monitoring of the captured audio, displaying a number of spectrum bars and a volume bar.
On the bottom you can see the range of frequencies analyzed by QLC+

2 Configuration

When double clicking on the widget in design mode, a panel is displayed, showing a complete set of options to tune the audio triggers functionality.
The first thing you can configure is the number of spectrum bars that you want to display and that you will need during your live show. The accepted number is between 5 and 32.
Once decided the number of bars, you can proceed to assign a functionality to each bar. There is a list showing the following options for the volume and spectrum bars:

Name	Can be "Volume Bar" or a spectrum bar shown as #number (start frequency - end frequency). Example: #5 (1250Hz - 1562Hz)>
Type	Indicates the type of functionality the bar will control. It can be: • None - No functionality assigned • DMX - Controls single DMX channels • Function - Controls a QLC+ function • VC Widget - Controls a Virtual Console widget (at the moment only buttons, sliders and tapping of speed dials)
Assign	When a type is selected, a button with ✓ icon is displayed to allow you to connect the bar to the desired functionality. Depending on which type you chose, clicking on this button will display the DMX channels, Functions or Virtual Console widgets selection dialogs.
Info	This column displays some additional information regarding the bar--functionality association. Depending on the type you chose, here you will find the number of DMX channels, the function name or the VC Widget name selected
Disable threshold	When selecting a Function or a VC Widget button, this column allows you to set a deactivation threshold percentage. When the spectrum or volume bar will go below this value, the associated function/VC button will be stopped/deactivated.

Enable threshold	When selecting a Function or a VC Widget button, this column allows you to set an activation threshold percentage. When the spectrum or volume bar will go above this value, the associated function/VC button will be started/activated.
Divisor	For speed dials, this will divide the taps: only every n-th tap will be actually sent to the speed dial. In other words, the tap happens only every n-th beat. If you want the tap every beat, enter 1, for every other beat enter 2. For once in a 4/4 measure, enter 4. For once in 3 4/4 measures, enter 12. Maximum is 64.

DMX channels

You can choose one or more channels of the currently patched fixtures. Those channels will be set proportionally to total volume, or volume in respective frequency band.

Functions

You can select one or more functions. Those functions are started when volume goes above Enable threshold in respective band, and stopped when it goes below Disable threshold.

VC Widgets

You can select only **ONE** widget here, either a button, a slider or a speed dial. Depending on the type:

- **Button** is pressed when volume goes above Enable threshold, and released when the volume is below Disable threshold (similar to functions).
- **Slider** is moved proportionally to the volume (similar to DMX channels). For slider the thresholds do not apply.
- **Speed dial** is tapped when the volume goes above Enable threshold, and won't be tapped again until the volume falls below Disable threshold. That means you probably have to set the threshold little closer to each other than for buttons.

Note: Controlling widgets works now only if the audio triggers widget was created **later** than the controlled widget.

Select Input Channel

Whenever an Input Channel needs to be selected, for example when manually attaching a channel from an Input Plugin to a Virtual Console widget, the Select Input Channel dialog is used.

The dialog is very straightforward; there is a list of input universes and each universe that has an Input Line attached to it, displays the line's (or the attached device's) name after the universe number. Under each operational universe, there is a list of channels provided by that universe. You can choose from these channels or select one manually.

If you have an Input Profile attached to the universe, you see the individual channel names and numbers for the device. If there isn't a Profile attached to an input universe, you will need to type the channel number manually, by double-clicking the item that tells you to do that and type in the channel number manually. See also the tutorial on input profiles

NOTE: All Virtual Console widgets that have been made *external-input-aware*, provide also an option to automatically detect the input channel so that you don't need do this manual selection.

Virtual Console Widget Styling & Placement

Each widget can be "tuned" in terms of colors, font, frame style, stacking order, size and placement to better suit your vision of the perfect lighting console. Most of these options, except for moving and resizing, are available in the Virtual Console **Edit** menu that is accessed from the menu bar as well as a popup menu by right clicking a widget you wish to adjust.

NOTE: Virtual Console widgets' styling and placement options are available only when the application is in Design Mode.

Moving Widgets

Widgets can be moved simply by dragging them as if you were dragging any icon on your desktop. Refer to your operating system manual on how to drag items.

To move a widget from one frame to another frame, you need to move the widgets through the Virtual Console clipboard; Cut the widgets from the first frame, then select another frame and paste the widgets to that frame.

Resizing Widgets

Each widget has a special resize handle ◢ on their lower-right corner that you can grab and then resize the widget simply by dragging the handle to an appropriate size.

Widget Background

To change a widget's background color, choose **Colour** from the **Background** menu and select a colour from the popup dialog. To set an image as the widget's background, choose **Image** from the **Background** menu and select an image file from the popup dialog.

To restore the default background color (and to clear an image), choose **Default** from the **Background** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of background colours or images. Try changing the desktop theme if you encounter this problem.

Widget Foreground

To change a widget's foregroung color, choose **Colour** from the **Foreground** menu and select a colour from the popup dialog.

Buttons allow an icon on the foreground, overriding any text set as their name. To change a button's icon, select **Choose...** from the **Icon** menu and select an image file from the popup dialog.

To restore the default background color (and to clear an icon if applicable), choose **Default** from the **Background** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of foreground colours or images. Try changing the desktop theme if you encounter this problem.

Widget Borders

To change the border style for a widget, you can choose either **Raised** or **Sunken** from the

Border menu. To completely remove the borders of a widget, choose **None** from the **Border** menu.

NOTE: Some operating systems and/or visual themes do not allow manual setting of borders. Try changing the desktop theme if you encounter this problem.

Fonts

You can change the font for all child widgets of a frame at the same time by changing the font of the frame itself. Alternatively, you can change the font for each widget separately. Fonts can be changed by selecting **Font...** from the **Font** menu. To restore the font back to the widget's parent frame's font, select **Default** from the **Font** menu.

Widget Stacking Order

In addition to the normal 2-dimensional vertical and horizontal directions, you can also adjust widgets' placement in the 3rd dimension; depth. If you see a widget going behind another control, you can **raise** the widget on top of all other widgets by selecting **Raise** from the **Stacking Order** menu. Likewise, if you wish to put a widget behind all other widgets, you can select **Lower** from the **Stacking Order** menu.

Simple Desk

The simple desk emulates a typical 12/24 lighting console that is able to control a full 512-channel DMX universe with a multiple cue stacks operated with playback sliders. The desk can be thought of as a desk within a desk since it operates separately from other QLC+ components. For example, cues within the cue stack are not visible in the Fixture Manager and Scenes are not visible in the cue stack.

Controls - Universe

The universe box contains the sliders that are used to control individual DMX channels in the first DMX universe. Since 512 sliders cannot nicely fit on the screen at the same time, they have been divided into pages. By default, each page contains 32 sliders but that can be tuned.

❐	Switch the view mode from all channels to fixture channels. Please note that if no fixtures has been defined, the second mode will result empty
⬅	Skip to the previous DMX page.
Universe page box	Displays the current DMX page. You can skip to a page by writing the page number directly onto this box or use the mouse wheel to skip between pages quickly.
➡	Skip to the next DMX page.
✖	Reset all DMX sliders back to zero.
GM	The Grand Master

Controls - Playback

The playback box contains a bunch of playback sliders that each may contain a cue stack. The playbacks can be used to "play back" the contents of their cue stacks and they can also be used to adjust their overall cue intensity.

✓	Select the currently-active playback, whose cue stack is shown on the right side of the screen.
Playback slider	Adjust the playback's cue stack intensity. When the slider is drawn all the way to zero, the cue stack is stopped. Any value above zero will start playing the playbacks' cue stack.
★	Flash the playback's first cue.

Controls - Cue Stack

The cue stack box shows the currently selected playback's cue stack contents.

◀	Skip to the previous cue (or start playback on the last cue in the cue stack with full intensity).
■	Stop the currently-active cue stack.
▶	Skip to the next cue (or start playback on the first cue in the cue stack with full intensity).
✎	Switch to/from cue edit mode. When this button is pressed, you can edit the contents of individual cues; the currently active cue's contents are reflected on the DMX sliders. The Fade In speed, Fade Out speed and Duration as well as the name of individual cues can be adjusted with the speed dials that are displayed in a separate tool window. You can also select multiple cues to adjust their speeds all at once, but then the DMX sliders are disabled to prevent you from accidentally overwriting all cue contents with the same values.
●	Record a new cue, taking its contents from the current DMX slider values.
Cue Stack	This box displays the currently selected playback's cue stack contents, along with the index number (1, 2, 3...), Fade In speed, Fade Out speed, Duration and an optional name for each cue. You can change the cue order by dragging the cues on top of each other, either one at a time or multiple cues at the same time.

Tuning

Please refer to the Manual parameters tuning Simple Desk section

Input/Output Mapping Howto

This howto document tells you how to patch plugins and their input/output lines (physical input/output devices) to QLC+'s universes.

QLC+ by default provides 4 input and 4 output universes. These numbers may be increased if needed, but a recompilation is necessary, since they are hardcoded (look in engine/src/doc.h).

Input/Output Manager

To access the Input/Output Manager, just click on the tab with the ⇄ icon placed on the bottom of the QLC+ main screen.
The screen is composed in this way:

- On the left side there is the list of internal universes that QLC+ can manage
- On the right side there is the list of devices and their mapped inputs, outputs and feedback lines that QLC+ has detected
- On the bottom right side there is a panel displaying brief information of the currently selected device

Every device has a checkbox whenever an input, output or feedback line is available.
Each QLC+ universe can map a single input, a single output and a single feedback line

Some plugins might require configuration before they can be used so you might not be able to see all inputs/outputs at first. The configuration button is place next to the information panel and it is enabled if the plugin allows any manual setting.

The button is icon is: 🛠

Patching

To patch a plugin's input/output line to the selected universe, you need to place a checkmark on that particular plugin's input/output line. You can have only one line assigned to a universe at a time, so when you check another line, the checkmark will move from its previous position to the one you checked.
If you don't see any line on a plugin, it means you don't have any device that QLC+ understands and you're left with the one and only (non-selectable) choice: None.

When an input/output line is checked, the corresponding universe information on the left side of the screen will change and will display the new configuration set.
The plugin information on the bottom right side of the screen will change as well and will give you the new status of the plugin line.

Input and Feedbacks

When a plugin input line is checked, it gets enabled right away, so you can perform a basic test to double check if your hardware is working properly with QLC+.

Just move a fader/knob on your external device, and if everything works fine, you will see a 🕹 icon appearing beside the corresponding universe on the left side of the screen.

If your input device supports a return channel, QLC+ can send a visual/mechanical feedback to it. Devices such as Behringer BCF2000 support this feature.

At the moment feedbacks are supported only through MIDI.

To learn how to setup your external input device for the best use with QLC+, please continue your reading with the howto for input profiles.

Input Profiles Howto

This howto document tells you how to associate input profiles to input universes and how to edit these profiles. You should first read the howto on input/output mapping so that you can access the input/output manager and know how to edit input universes.

Input Profile Manager

To access the Input/Output Manager, just click on the tab with the ⇄ icon placed on the bottom of the QLC+ main screen.
Select the desired universe on the left part of the screen and then click the Profile tab places beside the Mapping tab on the top right part of the screen.

Profile assignment

You will see a list of available input profile definitions, with a check mark on None. This means that the current universe doesn't have a profile assigned yet. To assign a profile to the universe, simply place a check mark on one of the available profiles by clicking on the empty check box. Only one profile can be assigned to one universe at a time, so the check mark will actually move from its previous position to the item that you have selected.

Add/Edit a Profile

It is probable that your input profile is not on the list and you need to create one of your own.

Click the ➕ create new input profile button to start making a profile definition for your input profile. Alternatively, you can edit any existing profile by selecting the appropriate item and clicking the ✎ edit button. The procedure is exactly the same in both cases from now on.

Input Profile Editor

A dialog is opened with entry fields for Manufacturer, Model and Type.

Enter the profile's manufacturer, the model and the input type to these fields and then click the Channels tab to edit the profile's channels.

Now you have two choices for adding channel definitions: Manual and Automatic:

Manual mode

➕ Click the add button to enter individual channel information by hand for each channel.

➖ Click to remove an existing channel

Automatic mode

🪄 Click the automatic wizard button to attempt automatic channel detection. You'll receive further instructions from QLC+. You must have an input plugin assigned to the current universe for this feature to work. Also, you must first stop the wizard to be able to navigate away from this dialog page.

Channel properties

When you add ➕ or edit ✏ a channel, a small window will be displayed, asking you to fill or change some parameters:

- **Number**: The channel number. Since QLC+ supports a wide variety of input plugins, the channel number might not be intuitive, so edit this if you know what you're doing.
- **Name**: The channel name. This is an arbitrary string to recognize the meaning of a channel.
- **Type**: The channel type. This can be: Slider, Knob, □ Button, ⬅ Previous page, ➡ Next Page, ★ Set Page
  Please note that the page change mapping must be used in conjunction with the Virtual Console Frame multi-page functionality.

Note that you cannot add the same channel multiple times to one profile.

Back to the input profile definitions panel

When you're done with channels mapping, click the OK button to accept changes and save the input profile. If you didn't enter a manufacturer/model for the profile, you'll be prompted to enter them before you can continue.
Now you should see the profile you just defined in the list of available input profiles. Remember how to assign it to the current universe? Scroll up to Profile assignment if you don't.

To remove any existing input profiles, click the ➖ remove button. Note that some profiles are so-called system profiles and cannot be removed unless you're the administrator.

That's all !
Now you can start using your preferred profile. When assigining an input channel to a QLC+ element (like Virtual Console sliders, channel groups, etc..) you will see that your Input profile mapping will be used.

ArtNet input/output plugin

Introduction

QLC+ supports the ArtNet protocol through an input/output plugin that receives and transmits packets on the network.
No extra requirement is needed, since QLC+ has a native implementation of the ArtNet protocol that works on Linux, Windows and OSX systems.

Configuration

The ArtNet plugin can be configured to send and receive packets from multiple network cards, virtual addresses and multiple universes per network card.
The desired configuration can be achieved quite easily by opening the plugin configuration panel.
The dialog will show all the possible choices depending on the network interfaces that QLC+ detects when opened.
The dialog will display two lists:

- The available networks and universes. Just click on the corresponding checkbox to enable the one you're going to use and it will be listed in QLC+ input/output manager.
  Networks and universes are displayed in the following form:
  - Network 1 - Universe 0
  - Network 1 - Universe 1
  - Network 1 - Universe 2
  - Network 1 - Universe 3
  - Network 2 - Universe 0
  - ...
  - Network 2 - Universe 3
  - ...
- A simple node inspector that will display the available ArtNet nodes discovered on the network

Compatibility

QLC+ has been tested with the following ArtNet softwares/devices:

- DMXking eDMX2 TX - Output device
- Enttec Open DMX Ethernet (ODE) - Output device
- OLA - Open Lighting Architecture - Input/Output software node
- ArtNet Controller LITE - Input controller for Android
- Controlador ArtNet DMX (Lite) - Input controller for Android

DMX USB input/output plugin

1 Introduction

The DMX USB plugin supports a variety of FTDI-based USB-to-DMX devices:

- DMXKing USB DMX512-A
- DMXKing ultraDMX micro
- DMXKing ultraDMX Pro
- Enttec DMXUSB Open
- Enttec DMXUSB Pro
- Enttec DMX USB PRO Mk2
- DMX4ALL USB-DMX STAGE-PROFI MK2
- FTDI USBCOM 485+

2 Configuration

DMX USB devices should be automatically detected from QLC+ and displayed in the input/output panels list.
If for some reason the auto-detection fails, you can "force" the type of your DMX USB adapter manually.

Click on the name of your device and open the configuration dialog by clicking on the 🛠 icon on the bottom-right side of the panel.
You will see a list of DMX USB devices currently connected to your computer. Each one has a drop down menu where you can force the device type.
Here's the meaning of each one:

- **Pro TX**: Enttec USB DMX Pro or most of the DMXKing devices in transmit (output) mode
- **Open TX**: Enttec USB DMX Open (output only)
- **Pro RX**: Enttec USB DMX Pro in receive (input) mode
- **Pro Mk2**: Enttec USB DMX Pro Mk2 - 2 outputs and 1 input available
- **Ultra Pro Tx**: DMXKing ultraDMX Pro output.
- **DMX4ALL**: DMX4ALL USB-DMX STAGE-PROFI MK2

3 Requirements

2.1 Linux

On all Linux distributions, you need to install libftdi. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

2.2 Mac OS X

On the Apple Mac OS X, you don't need any drivers at all since QLC+ uses the OS X native USB interface. Installing the D2XX drivers should cause no harm, but **DO NOT INSTALL VCP drivers** as they will definitely interfere with QLC+. If you have previously installed the VCP drivers, consult the FTDI installation guides on how to uninstall them.

2.3 Windows

On Microsoft Windows, the plugin needs the latest D2XX drivers from FTDI. Consult the FTDI installation guides on how to install the drivers. **DO NOT INSTALL VCP drivers** as they will probably interfere with the D2XX interface.

4 Tuning

Note: Manual tuning should never be performed except for some very particular cases. Use it at your own risk !
It is possible to change the DMX frame frequency for Enttec Open (and like) devices with a hidden settings key on each platform. The key tells QLC+ how many times each DMX frame (512 channels) should be sent to the universe per second. A value of "30" means 30 times per second (30Hz).
Please refer to the Manual parameters tuning DMX USB Enttec Open section

MIDI input/output plugin

Introduction

This plugin gives input/output support for the MIDI protocol and leave the user the freedom to control typical parameters like channels, Program Change, Control Change.
The MIDI plugin can be quite powerful used in combination with MIDI devices such as keyboards, MIDI controllers (like Behringer BCF2000 or KORG nanoKONTROL) or software audio sequencer like Cubase or Ardour 3.
The usage can vary from fader-to-fader controlling (the BCF2000 case) to sequence triggering for synchronized shows (metronome controlled gigs using an audio sequencer)

Configuration

When clicking on the configuration button, a window will appear, showing all the MIDI input and output lines detected.
Each line has two options that can be changed depending on your needs:

- **MIDI Channel**: This is the channel where QLC+ will receive or send data through the MIDI system. MIDI channels can go from 1 to 16. The special "1-16" channel will tell QLC+ to receive or send data on any MIDI channel.
- **Mode**: This is the MIDI mode that QLC+ will use to receive/send data through the MIDI system. This parameter can assume two possible values:
    - **Note velocity**: in this mode, QLC+ will receive/send data using MIDI notes velocity. MIDI notes can go from 21 (A0) to 108 (C8) and each note can have a velocity going from 0 to 127, which will be doubled inside QLC+ to fit the DMX range (0-255).
    - **Control Change**: this is one of the MIDI protocol messages (like Program Change) frequently used by MIDI controllers. Each device should list the supported CC messages in its user manual, so please consult it before using this mode. CC range can go from 0 to 127 and can carry values from 0 to 127, which will be doubled inside QLC+ to fit the DMX range (0-255).
    - **Program Change**: this is one of the MIDI protocol messages (like Control Change) frequently used by MIDI controllers. Each device should list the supported PC messages in its user manual, so please consult it before using this mode. PC range can go from 0 to 127 and can carry values from 0 to 127, which will be doubled inside QLC+ to fit the DMX range (0-255).

Feedbacks

The MIDI plugin is the one of the QLC+ plugins supporting feedbacks. When QLC+ detects a MIDI device with an output line, it will enable the feedback check box in the Input/Output panel.
Please note that output and feedbacks are exclusive, so they cannot be used both at the same time.
If your MIDI device supports a return channel, QLC+ can send a visual/mechanical feedback to it. Devices such as Behringer BCF2000 support this feature. This is very useful during live shows, to have an immediate feeling of the current state of faders mapped in QLC+.

A small trick that can be achieved with QLC+, is to use feedbacks as a generic MIDI output line to trigger external controllers/sequencers.
Let's make some examples:

- Input: **OSC** ---> Output: **DMX USB** --> Feedback: **MIDI**
- Input: **Enttec Wing** --> Output: **ArtNet** --> Feedback: **MIDI**

MIDI beat clock

Starting from version 4.5.0, QLC+ supports MIDI beat clock
Not to be confused with the MIDI timecode, the MIDI beat clock is a useful signal to sync BPM-based devices such as a drum machine with your lights controlled by QLC+.
Two special MIDI channels have been mapped in QLC+ to control your Virtual Console widgets with a beat clock.
Here's a brief explanation of the special channels:

- **Channel 530**: A signal is sent on this channel when a beat clock starts or stops.
- **Channel 531**: This signal is sent every BPM. QLC+ doesn't consider measures (e.g. 3/4, 4/4, 7/8), so you need to setup your MIDI clock considering how QLC+ will handle it.

Hint: If your controller is set to work at high BPM (e.g. 180-200), you might find difficult to catch the start signal. A trick to do this, is to catch the stop signal. Example:

1. Enable the QLC+ Virtual Console widget auto-detection
2. Hit play on your device generating the MIDI beat clock. QLC+ will detect channel 530 and will switch very quickly to 531
3. Stop the playback on your MIDI beat clock device. QLC+ will detect channel 530 again.
4. Disable the QLC+ Virtual Console widget auto-detection

In a similar way you can catch the beat signal as well. Just disable the auto-detection process before stopping the playback on your beat controller. (invert steps 3 and 4)

OLA output plugin

Introduction

The OLA plugin allows direct communication between QLC+ and the OLA framework on the same machine.

Requirements

The OLA plugin requires OLA to be installed on the system.
Since OLA doesn't run on Windows, only Linux and OSX users can benefit from this plugin.
Information on how to download and install OLA can be found here.
QLC+ needs the OLA server to be running to be able to communicate with the OLA framework.
This can be done either manually by starting up "olad" from a terminal or in the configuration panel, by ticking "Run standalone OLA daemon".

Configuration

When pressing the configuration button over a OLA output line, a small popup window will appear, showing the basic information of how QLC+ outputs are mapped against OLA universes.
On the bottom, a check button will allow you to force the OLA server startup.

OLA Setup

When you make sure everything is working in QLC+ and how universes are mapped, you can setup OLA to output the signal received from QLC+ to a DMX device, either USB or over the network.
Here's an introduction of OLA usage.
Basically you need to open a web browser, connect to http://localhost:9090 or http://127.0.0.1:9090 and add a universe with the same number mapped in QLC+ and select the desired output line.

OSC input/output plugin

Introduction

QLC+ supports the OSC protocol through an input plugin that receives packets from the network.

Requirements

The OSC protocol is available thanks to the liblo library, which is linked dynamically on Linux systems and statically on Windows and OSX.
In other words, Windows and OSX don't need any extra software, while Linux systems need the liblo library installed.

Configuration

The OSC plugin supports up to 4 inputs, one for each QLC+ universe. Each input has to listen to a unique port and this can be configured from the plugin configuration panel.
For each OSC network the following parameters can be set:

- **Port:** The port QLC+ will listen to receive OSC data from your external controller (Just enter the same port your external OSC controller is using)
- **Output address:** In case you need to output DMX data through the OSC plugin (either for feedbacks), enter here the address to send to. This must be specified in the form: *IPAddress:port*

Controllers

QLC+ has been tested with the following OSC controllers:

- TouchOSC. A pre-defined input profile is ready to use for the Mix16 layout.

Peperoni output plugin

1 Introduction

The Peperoni Output plugin supports USB-DMX output devices produced by Peperoni Light.

2 Requirements

2.1 Linux

On all Linux distributions, you need to install libusb. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

2.2 Mac OS X

Nothing special is needed for Mac OS X. All required components are already inside the Q Light Controller Plus application bundle because QLC+ uses the OS X native USB interface.

2.3 Windows

You must install the Peperoni USBDMX Windows drivers. Usually these come on a CD along with the Peperoni device you've bought.
If after installing the driver, QLC+ still doesn't detect your Peperoni device, copy the usbdmx.dll file you find inside the driver ZIP package (i386 folder) into the main QLC+ folder.

- Unzip the driver package to a folder on your hard disk.
- Plug the peperoni device to a USB port.
- If you're running a 32bit version of Windows, point the "Found new hardware" wizard to look for the driver under **windows/i386**.
- If you're running a 64bit version of Windows, point the "Found new hardware" wizard to look for the driver under **windows/ia64** or **windows/amd64**.

uDMX output plugin

1 Introduction

The uDMX Output plugin supports the Anyma uDMX USB-DMX interface on Linux and Mac OS X.

2 Requirements

2.1 Linux

On all Linux distributions, you need to install libusb. If you install QLC+ with the Ubuntu Software Center or some other automatic installer tool, this library will be installed automatically for you.

2.2 Mac OS X

On the Apple Mac OS X, you don't need any drivers at all since QLC+ uses the OS X native USB interface.

2.3 Windows

This device is not yet supported on Windows.

3 Tuning

It is possible to change the DMX frame frequency for all uDMX devices with a hidden settings key on each platform. The key tells QLC+ how many times each DMX frame (512 channels) should be sent to the universe per second. A value of "30" means 30 times per second (30Hz). Please refer to the Manual parameters tuning uDMX section

Velleman output plugin

1 Introduction

The Velleman Output plugin supports the Velleman K8062D interface on Windows operating systems.

2 Requirements

2.1 Windows

You must install the proprietary Velleman K8062D library to **C:\Windows\System32**.

Fixture Definition Editor

Fixture Definition Editor is a separate application for creating and modifying fixture definitions used by QLC+. The definitions tell QLC+ (and users) important details about fixtures, such as which channel is used for pan movement, what value in which channel changes the beam colour green, how the fixture is reset etc...

The main window in the Fixture Editor is just an empty workspace that contains the actual editor windows used to actually edit fixture definitions.

Controls

	Create a new fixture definition. Opens an empty Fixture Editor window.
	Open an existing fixture definiton. Open the fixture definition in a Fixture Editor window.
	Save the fixture definition in the currently active Fixture Editor window.
	Save the fixture definition with a given name in the currently active Fixture Editor window.

Capability Editor

With the Capability Editor you can define a value range for a certain functionality provided by a fixture channel. For example, if values 0-15 produce a white colour capability, you might put:

- 0 to the **Min** field
- 15 to the **Max** field, and
- "White" to the **Description** field

Controls

Min	The minimum value of the capability
Max	The maximum value of the capability
Description	The name or description of the capability. Don't be too elaborate, just a few words.

Click & Go capability

The Click & Go functionality for a channel starts from the Capability Editor.
When a channels group is set to Gobo, Colour or Effect, it is possible to assign the color or a picture that will be displayed by Click & Go on Scene Editor, Simple Desk and Virtual Console Slider.
This allows a very quick visual feedback, and will avoid you to remember the meaning of each range of values.
When a channel can support Click & Go, some additional controls will appear on the right side of the Capability Editor, displaying 3 butons and a preview box. Here's the meaning of each one:

Picture ...	When clicking on this button you can select a picture for this range of values, to be displayed by Click & Go for visual association and quick access
Colour 🎨	When clicking on this button you can select a primary colour for this range of values. When a primary colour is set, the 'Second colour' button will be enabled. See below.
Second colour 🎨	When clicking on this button you can select a secondary colour for those cases where a fixture supports it. For example this is quite useful for those moving heads with colour wheels supporting intermediate positions.

Capability Wizard

Capability Wizard is a handy tool for creating multiple capability value ranges of same size. Usually this applies to fixed colours, gobo indices and various macro channels.

Controls

Start	The starting value for new capabilities. Sometimes there might be other capabilities at the start of the channel's value range that you can skip by adjusting this value.
Gap	The size of each value range.
Amount	Number of capabilities to create.
Name	The common name for each capability. You can use the hash mark # to denote a place for an index number (i.e. "Gobo #" creates Gobo 1, Gobo 2, Gobo 3...)
Sample	Every time you change a parameter in the wizard, this list is updated to show you a sample of what kinds of capabilities will be created once you click OK.

Channel Editor

The Channel Editor is used to edit individual channels and the value ranges of each capability (a green colour, a certain gobo, prism rotation, etc..) that a fixture channel provides. Refer to your fixture's manual to get a detailed list of the fixture's channels and values.

Controls

Name	The name of the channel
Group	The channel's logical group (role in the fixture). • Assign a channel to the **Colour** group if the channel controls a fixed colour wheel or pre-defined colour macros. *Don't* assign individual RGB/CMY colour channels to the Colour group but instead use the **Intensity** group as described below. The channel obeys the LTP rule. • Assign a channel to the **Gobo** group if the channel controls gobo wheel position or indexing. The channel obeys the LTP rule. • Assign a channel to the **Speed** group if the channel controls something related to speed (gobo rotation, rainbow speed, tracking speed). The channel obeys the LTP rule. • Assign a channel to the **Prism** group if the channel controls a prism. The channel obeys the LTP rule. • Assign a channel to the **Prism** group if the channel controls a prism. The channel obeys the LTP rule. • Assign a channel to the **Shutter** group if the channel controls a shutter or strobe. The channel obeys the LTP rule. • Assign a channel to the **Beam** group if the channel controls a beam shaper. The channel obeys the LTP rule. • Assign a channel to the **Effect** group if the channel controls something that doesn't quite fit into any of the other groups. The channel obeys the LTP rule. • Assign a channel to the **Maintenance** group if the channel contains fixture resetting or cooling fan controls or something similar. The channel obeys the LTP rule. • Assign a channel to the **Nothing** group if the channel should not appear in the Scene Editor or Fixture Console. • Assign pan channel(s) to the **Pan** group and respectively, tilt channel(s) to the **Tilt** group. See the **Control Byte** property below. The channel obeys the LTP rule. • Assign a channel to the **Intensity** group if the channel should obey the HTP rule. Also, the Grand Master controls only Intensity channels by default. Note also when you create RGB/CMY channels, that they should be assigned to the *Intensity* group, and **NOT the Colour group**. The Colour Tool in Scene Editor is

	available only if a fixture provides **Intensity** channels for RGB/CMY, whose **Colour** properties have been set accordingly (see the **Colour** setting below).
Colour	Only available when the channel has been assigned to the **Intensity** group. Tells, whether the channel controls a specific colour out of the primary colours **Red, Green, Blue** or **Cyan, Magenta, Yellow**. If the channel controls the fixture's master intensity, the **Generic** colour setting should be used.
Control Byte	Only available when the channle has been assigned either to the **Pan** or **Tilt** group. If the fixture supports only 8bit movement (only one channel for each), assign the **Coarse MSB** control byte to the both of them. If, however, the fixture supports 16bit movement (two channels for each), you should assign the **Coarse MSB** byte to the channels that provide coarse head/mirror movement and the **Fine LSB** byte to the channels that provide subtle head/mirror adjustment.
Capability list	Shows you a list of value ranges for the currently edited channel. If a channel provides only one capability (for example pan or dimmer) the only thing needed in the list is one capability with a range of 0 to 255. For more elaborate capabilities, such as colours or gobos, you should create capability ranges for each of the colours (for example 0-15 white, 16-32 blue...)
✚	Add a new capability to the channel, using the Capability Editor.
▬	Remove the selected capabilities from the channel.
✎	Edit the selected capability, using the Capability Editor.
🪄	Create new capabilities quickly with the Capability Wizard.

Fixture Editor

Fixture Editor windows contain everything needed to edit one Fixture Definition at a time. The windows are separated in three tabs: **General**, **Channels** and **Modes**.

Controls - General Tab

Manufacturer	The fixture's manufacturer name. For example "FooCompany".
Model	The fixture's model name. For example "FooZapper 2000".
Type	The fixture's generic type.

Controls - Channels Tab

The channels tab contains all possible channels that the fixture understands in all of its modes. The channel order doesn't matter in this tab at all. Instead, channels are arranged in certain order in each mode in the **Mode** tab. On the **Channel** tab, only the channel names, their **capabilities** (i.e. value ranges and their purpose) matter.

Channel list	Shows all of the channels that the fixture understand in all of its modes. Each channel item also shows all of its capabilities in a sub tree.
➕	Add a new channel to the fixture using the Channel Editor.
➖	Remove the selected channels from the fixture **and from all modes**.
✏	Edit the currently selected channel using the Channel Editor
⧉	Copy the currently selected channel to the clipboard. Channels in the clipboard can also be pasted to other fixture definition windows.
📋	Paste a channel from clipboard to the fixture definition. Channels in the clipboard can also be pasted to other fixture definition windows.
✓	Open or close all of the channel nodes in the channel list.

Controls - Modes Tab

The modes tab contains all modes the fixture can be configured to.

Mode list	Displays all modes for the currently edited fixture. Each mode item can be opened to display the set and order of channels in that mode. • Name: The name of the mode (each name must be unique) • Channels: Number of channels in each mode
➕	Create a new mode for the fixture, using the Mode Editor.

−	Remove the currently selected mode from the fixture. Removing a mode does not destroy any channels or other modes.
✎	Edit the currently selected mode, using the Mode Editor.
⧉	Create a copy of the currently selected mode to the same fixture. Since modes are tightly coupled to a certain fixture's channels, modes cannot be copied across fixtures.
✓	Open or close all mode items.

Mode Editor

The Mode Editor is used to create and edit modes by picking sets of **Channels** in certain order (as defined by the fixture's manufacturer). Each editor window is divided into three tabs: Channels, Heads and Physical.

Controls - Channels Tab

In the Channels tab you can place the fixture's channels in some order to form an actual representation of DMX channels that the fixture understands when it has been configured in that particular mode.

Mode Name	The name of the mode (must be unique for each mode).
Channel list	Displays all channels present in the current mode, in their proper order.
➕	Add channels from the fixture's channel collection to the mode. You can create/edit channels in the Fixture Editor's **Channels** tab. When clicking on this button, a new window will be displayed, showing two lists: • The list on the left shows the channels that have not been added to this mode yet. If you're creating a new mode, this list will show all the channels you created in the Fixture Editor's channels tab • The list on the right shows the channels that define the mode you're editing. If you're creating a new mode, this list will be empty. Please note that the order of the channels in this list is fundamental for the fixture mode definition. To move items from a list to another either use the central buttons or simply drag and drop them. When done, press OK to update the mode channel list.
➖	Remove the selected channels from the mode. Other modes' channel selections are left untouched.
⬆	Raise the selected channel up by one.
⬇	Lower the selected channel down by one.

Controls - Heads Tab

In the Heads tab you can define multiple heads to a single fixture, so that QLC+ knows to treat each of them individually in certain situations (such as Fixture Groups). If a fixture contains only one head when it's configured to use the currently edited mode (i.e. all of the mode's channels control one head at a time) there is no need to define the head. If, however, the fixture has multiple heads that you wish to be able to control, you must define each head here.

Heads don't have names as they can be thought of as simple "sub-fixtures" inside a fixture. Instead, they are automatically given index numbers. Heads' ordering should follow the physical fixture's configuration as closely as possible. So, if the heads go 1, 2, 3, 4 in the real world, don't define them as 4, 2, 1, 3 or something likewise annoying.

Head list	Displays the list of heads currently defined for the fixture.
➕	Add a new head to the fixture, using the Head Editor.
➖	Remove the selected head from the fixture.
✎	Edit the selected head using the Head Editor.
⬆	Raise the selected head upwards by one. You can use this to change the order of the heads within the fixture.
⬇	Lower the selected head downwards by one. You can use this to change the order of the heads within the fixture.

Controls - Physical Tab

Because some fixtures allow their physical properties (e.g. pan/tilt extent) to be altered between modes, it is beneficial that also the fixture's actual physical properties are tied to fixture modes.

Bulb	• Type: The type of the actual light source within the fixture. • Lumens: The light source's total luminous output in lumens. • Colour Temperature (K): The light source's colour temperature in Kelvins
Lens	• Name: The type/name of the lens, if applicable. • Min. degrees: The fixture's minimum beam angle in degrees. • Max. degrees: The fixture's maximum beam angle in degrees.
Technical	• Power Consumption: The fixture's total power consumption in Watts. • DMX Connector: The type of the fixture's DMX connector.
Dimensions	• Weight: The fixture's total weight in Kilograms. • Width: The fixture's total horizontal width in millimeters. • Height: The fixture's total vertical height in millimeters. • Depth: The fixture's total depth in millimeters.
Focus	• Type: The method of focusing the beam to a certain point. For stationary fixtures this can be set to Fixed. • Pan Max Degrees: The maximum pan width in degrees. • Tilt Max Degrees: The maximum tilt height in degrees.

Head Editor

The Head Editor is used to create and edit heads by picking sets of **Channels** that are dedicated to a single head (as defined by the fixture's manufacturer).

Controls

The controls are very simple: place a checkmark on each channel that is used ONLY by the head you are currently editing. To add a head, see Mode Editor.

Channel list	Displays the list of all fixture channels available in the current mode. Channels that have been assigned to another head are disabled and cannot be selected because each channel can belong to one head at a time.

Scene tutorial

This tutorial covers just the basic chain of creating a fixture, a scene and a button and how to make them work. The point is to give the user a quick yet complete here-hold-my-hand-experience on how the Q Light Controller Plus is thought to work as a software for DMX lighting automation.

Before you start reading this tutorial, please familiarize yourself with the Main window and its parts.

Now we can really start!

Fixture Manager

Open the fixture manager panel now by clicking its icon.

The fixture manager is the heart of QLC+ fixture-oriented architecture. As its name already says, you can manage (add, remove and edit) your lighting fixtures from the fixture manager. On the left side of the manager there is a list that contains all of the fixtures in the current workspace (it will be empty until you have added some fixtures). On the right side you can see some common information related to the currently selected fixture. At the top of the fixture manager there is again another toolbar, containing the following buttons (from left to right):

Add new fixtures

Remove selected fixtures

Configure the selected fixture

Group a fixture selection

Ungroup a fixture from a group

Import a previously saved list of fixtures

Export a list of fixtures

Adding a fixture

Add a fixture to the workspace by clicking the add button.

On the left side of the dialog you can see a list of available fixture manufacturers. Each manufacturer item is actually a folder containing a number of fixture models produced by the manufacturer. You can find for example a "DJScan250" under the "Futurelight" folder. As you click a fixture from the list, you can see the Channels field on the right side change to display the number of DMX channels required by the selected fixture. There is also a list of the fixture's channels just under the Channels box.

Go ahead and select Futurelight DJScan250 but don't click OK just yet.

You can edit the new fixture's name in the Name field or you can stick to the default that the application suggests. If the fixture has different operational modes (different sets of channels), you can select one from the Mode box. The DJScan250 doesn't have more than one, so there's just "Mode 1" there. The fixture's DMX address can be set in the Address field and should be the same as the actual physical fixture's DMX address. The Universe field is used to assign the fixture to a physical DMX output universe. Usually each universe has its own cable coming from the PC.

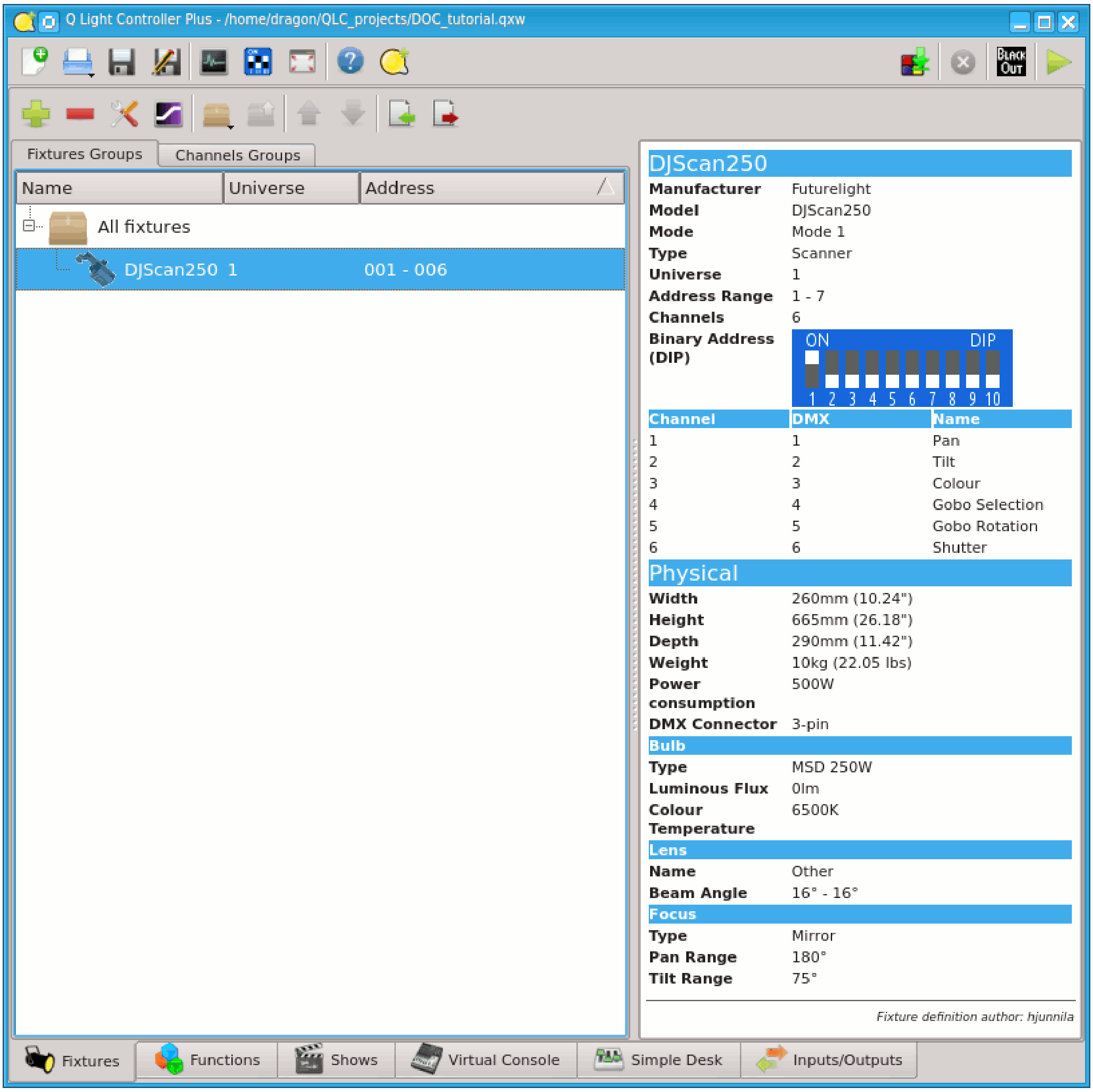
If you wish to add multiple fixtures of the same type, you can increase the value in the Amount box. If you wish to leave a gap between each fixture's address space, you can change the value in the Address gap box. **Let's leave these be for now.**

If you don't understand the DMX addressing principles, please consult your lighting equipment manuals for more information. In short, a DMX address is the first DMX channel of a fixture. In the case of a DJScan250 (which uses 6 channels), assigning for example 1 as its DMX address, reserves channels 1, 2, 3, 4, 5, and 6 for the fixture. The next fixture must then be assigned to DMX address 7 to prevent channel overlapping. **Let's use address 1 now.**

Back to the fixture manager

Click OK to close the dialog and add one Futurelight DJScan250 to the workspace.

On the left side of the fixture manager you can now see the fixture that we just added. On the right side, you can see the fixture's information. You can edit the fixture's name, address and universe by clicking the Configure button. You can also change the fixture definition thru the configuration dialog, but since we're quite happy with the fixture's information, we don't go there.



Name	Universe	Address
All fixtures		
DJScan250	1	001 - 006

DJScan250

Manufacturer	Futurelight
Model	DJScan250
Mode	Mode 1
Type	Scanner
Universe	1
Address Range	1 - 7
Channels	6
Binary Address (DIP)	

Channel	DMX	Name
1	1	Pan
2	2	Tilt
3	3	Colour
4	4	Gobo Selection
5	5	Gobo Rotation
6	6	Shutter

Physical

Width	260mm (10.24")
Height	665mm (26.18")
Depth	290mm (11.42")
Weight	10kg (22.05 lbs)
Power consumption	500W
DMX Connector	3-pin
Bulb	
Type	MSD 250W
Luminous Flux	0lm
Colour Temperature	6500K
Lens	
Name	Other
Beam Angle	16° - 16°
Focus	
Type	Mirror
Pan Range	180°
Tilt Range	75°

Fixture definition author: hjunnila

My first function

Open the function manager by clicking its button on the main toolbar

If the fixture manager was the heart, function manager must be the brain of QLC+. With it, you can add, remove and edit various functions that perform the actual light automation for you. Let's look at the view first. There's again a toolbar containing the following icons (from left to right):

Add a scene

Add a chaser

Add a sequence

Add an EFX

Add a collection

Add a RGB matrix

Add a script

Open the Functions wizard
Copy the selected functions
Destroy the selected functions

Below the toolbar, there is a list of functions within the current workspace. Since we haven't yet created any, it's very much empty.

Scene editor

Add a new scene to the workspace by clicking the scene button

With the scene editor, you can create scene functions that basically contain values for a number of channels that relate to certain fixtures. On the left side of the editor there is a list of fixtures used in the scene, which is empty (but not for long). There are also some buttons to add/remove fixtures and enable/disable all the selected fixtures channels.
On the right side of the editor the is a list of channel groups used in the scene. This will not be used in this tutorial.

Add a fixture to the scene by clicking the ✚ add button. Select our one and only fixture, the DJScan250 from the list that pops up and click the OK button.

Now the scene has one fixture to control. Notice also that now there is new tab just beside the General tab with our fixture name. **Click the tab that says DJScan250.**

Now you can see a panel with 6 sliders & buttons, each of them representing a channel within the fixture, but all of the channels are disabled. Above each of the buttons (the ones with icons) there is yet another box which, when checked, enables the corresponding channel in the scene. If a box is not checked, that channel will not be touched by the scene at all. This is very useful when you wish to create for example a function that just sets the color of a scanner, without touching the gobo, intensity, pan, tilt and other features that you might wish to stay the way they are.

Set channels 3, 4 and 6 enabled by clicking on their check boxes.

You'll see that channels 3 and 4 changed their appearance and are no longer grayed out. You can also move their sliders and click their buttons. When you click on a button, you get a list of available capabilities that the fixture can do when a certain value is set to the channel. In the case of a DJScan250, channel 3 controls the color wheel, channel 4 controls gobo selection and channel 6 controls the fixture's shutter.

If you click the button on channel 3, you get a list of available colors that the fixture supports. Since many of these capabilities are specified by the manufacturer as a value range rather than a single value, many of these capabilities contain yet another sub-menu. Let's try setting a color.

Click the color button on channel 3, then move your cursor to "Orange" and select "80" from the sub-menu by clicking on that value.

Notice that the slider on channel 3 also moved up and the value above it now shows 80. If you have already patched an output plugin to the first universe, you may already see some action going on in your DMX equipment, since the scene editor also sends real DMX data to your fixtures as you edit the values. If you haven't done any mappings, don't feel bad, we'll get to it.

Next, click the ▲ gobo button and choose "Gobo 7", value 126 and then click the intensity button and choose "Shutter open", value 255.

Now we have a scene function that sets the value of DJScan250's channel 3 to 80, channel 4 to 126 and channel 6 to 255. Now all we need is a nice and descriptive name for the scene. Click the General tab to go to the general page where we started with the scene editor.

You can set a nice name to the scene by writing it to the Scene name edit box. **Type: "*DJScan250 Orange Gobo 7*" there.**

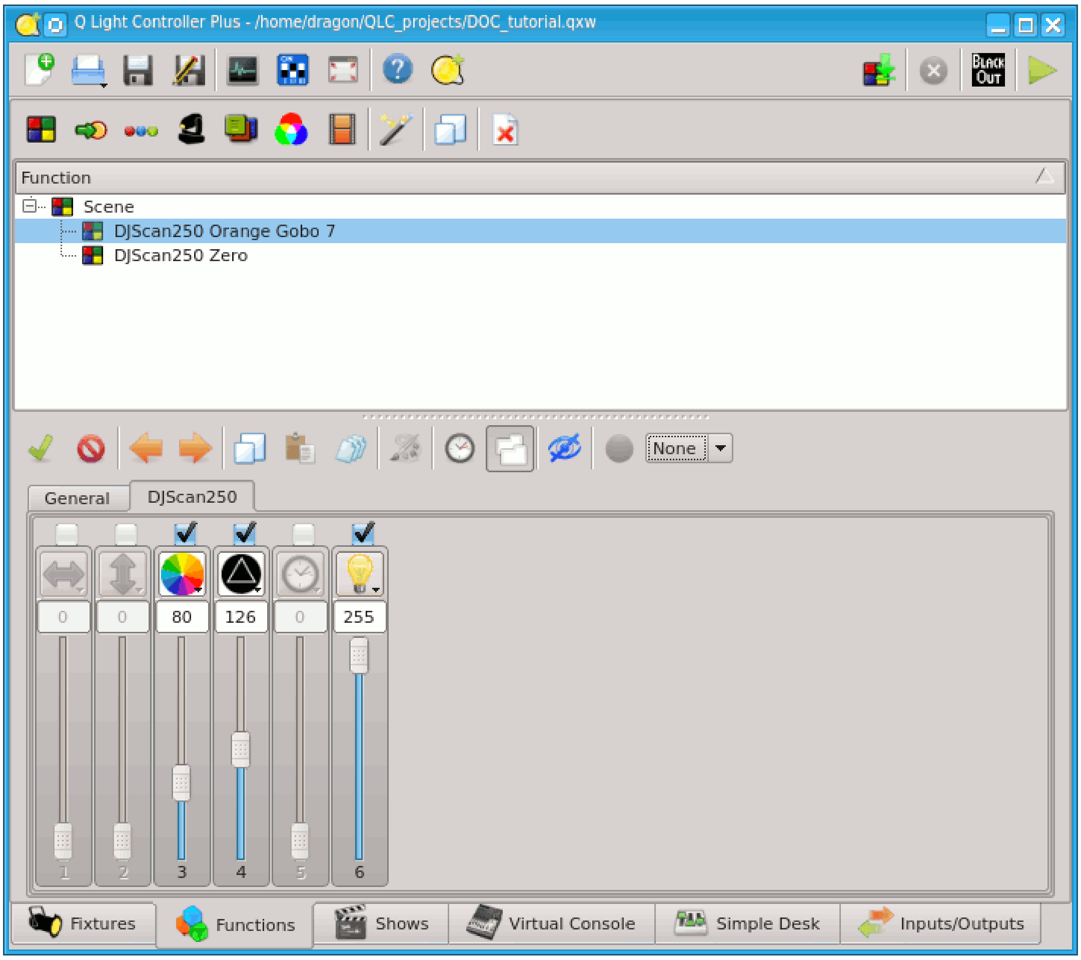
Now we're going to set a Fade In time to the Scene, so that when we play it it will fade to the values we set in a given amount of time.
Click on the 🕒 icon from the Scene Editor toolbar. A tool will be displayed, allowing you to choose the Fade In and Fade Out times of the Scene. Let's change Fade In to 5 seconds. Either use the speed dial widget or manually write '5' on the second last text field where 0s is written.
Close the Fade tool by clicking again on 🕒.

Back in the function manager

You can now see that the function manager displays a function called "DJScan250 Orange Gobo 7".
If you click the right mouse button over a function item, you get the same menu functionalities that are available in the upper part of the function manager. You can add new functions or edit existing ones. **But let's not edit this function anymore.**

Another function

Create another function just like the one you just made, but set the values for channels 3, 4 and 6 to 0 and name the function "DJScan250 Zero".



Virtual Console

Speaking of vital organs, we have already covered the heart and brain of QLC+, and we're only missing the body with its limbs to make the whole pack work. Well, so much for ridiculous analogies, let's move on and make our *"DJScan250"* fixture and its *"DJScan Orange Gobo 7"* function do some actual work for us.

You can close the function manager and the fixture manager now, if you want to make up some room but it's not necessary.

Click the virtual console button on the main window to show the virtual console tab.

Creating a button

At first, the virtual console is just an empty window without much to look at. There's a toolbar at the top of the panel, with icons to Add new widgets, Edit for editing widget properties and Tools for various tools to control the virtual console behaviour. You can also click the right mouse button on any virtual console widget to access a menu that contains most (but not all) of the options that are accessible thru the menu bar.

Click the □ icon to add a new Button widget.

An empty button appears to the virtual console.

Attaching a function to the button

Double click on the button or on the ✎ icon to bring up the button's configuration dialog.

From this dialog you can edit the button's properties:

* Set the Button label that is displayed on the button
* Attach a Function to the button
* Bind a Key combination to act as button presses * Bind the button to an External Input source
* Set the Button press behaviour

Click the attach button to open a function selection dialog. Double-click on the *"DJScan250 Orange Gobo 7"* function to attach it to this button.

We don't necessarily need a name for the button, but if you feel like giving it one, please do.

Click OK to accept these changes and close the dialog.

A little fine-tuning

If you gave the button some name, you'll notice that it doesn't quite fit and gets trimmed to a rather short version of the original (unless you gave it a two-letter name). You can resize the button to any size you like by grabbing it from the box on the button's lower-right-hand corner and dragging the button a little bigger. But hey, let's change the button's color now.

Click on the button again, and then click on the icon. Select an orange-tinted color from this dialog and click OK.

Now the button should have an orange background color. Move the button a bit to the side so that the next button won't appear on top of it. Well, there's no harm in that, it's just an inconvenience -- you want to be able to see both buttons, do you not?

Another button

Create another button just like the first one, but attach the function *"DJScan250 Zero"* to the second button and set the button background color to black and foreground (text) color to white.

Seeing the results

Since we haven't covered DMX output patching at all, you probably have a no output plugin assigned to all output universes and you can't get any real DMX out from your computer. This is OK for now. If you're interested in output mapping details, refer to the Output mapping howto.

Click the monitor button on the main application toolbar to bring up the DMX monitor window up.

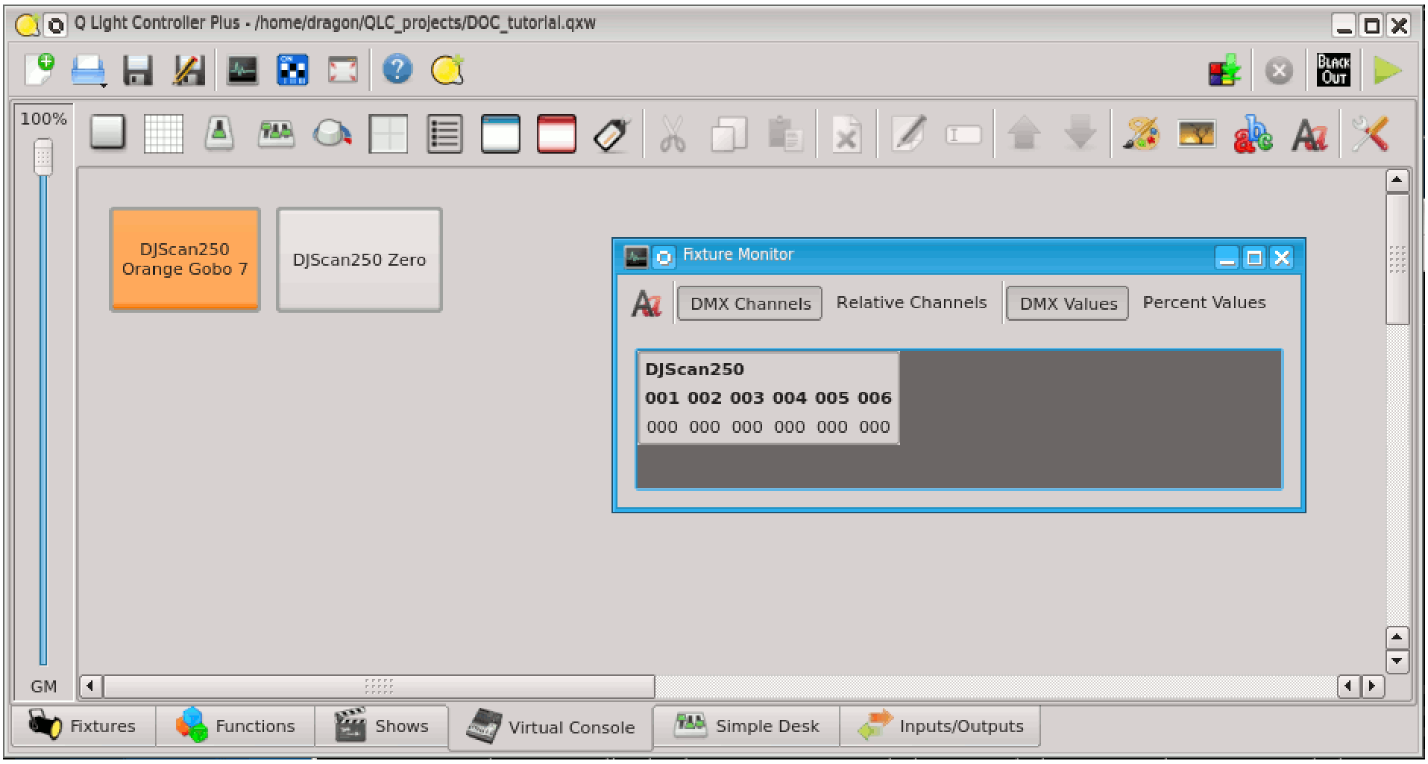
You should see a bunch of numbers, and the name of our fixture "DJScan250" inside a bar over numbers 001 - 006. These numbers represent DMX channels and the values below them represent these channels' values. Since we're operating on the dummy output plugin, the monitor is all that we see for now.

Click the ▶ mode switch button on the right-hand corner of the main toolbar to switch to operate mode.

Hold your breath...

Pay close attention to the monitor while you click the buttons on the virtual console (you know, the one we just created). Do you see some running numbers that gradually go towards 80 on channel 3, 126 on channel 4 and 255 on channel 6? Nice.

Note that if you click both buttons simultaneously, the result is usually far from what is wanted. You need to stop the previous function by clicking its button once more (so that the button flashes and stays up) to stop the function and then start the other function.



```
DJScan250
001 002 003 004 005 006
000  000  000  000  000  000
```

Command-Line Parameters

-c or --closebutton

Description: Define a place for a close button in virtual console. The button can be used to close QLC+. Has effect only in kiosk mode (see -k) and is most useful when in fullscreen mode without a window manager.

Parameters: x,y,w,h

Examples:
Create a button at (x400, y500) whose size is (w70, h50):
qlcplus -c 400,500,70,50
qlcplus --closebutton 400,500,70,50

-d or --debug

Description: Enable debug mode and optionally set the output level.

Parameters: Level (see QtMsgType)

Examples:
Enable debug mode and set debug level to 0:
qlcplus -d
qlcplus --debug

Enable debug mode and set debug level to 3:
qlcplus -d 3

-f or --fullscreen

Description: Start the application in fullscreen mode

Parameters: Method (either 'normal' or 'resize')

Examples:
Tell the window manager to give the whole screen space to QLC+:
qlcplus -f
qlcplus --fullscreen
qlcplus -f normal
qlcplus --fullscreen normal

Resize QLC+ to take up the whole screen space (useful in custom X11 sessions without a window manager):
qlcplus -f resize
qlcplus --fullscreen resize

-h or --help

Description: Display command-line help (only in Linux & OS X)

Parameters: None

Examples:
Display the command-line help:

```
qlcplus -h
qlcplus --help
```

-k or --kiosk

Description: Enable kiosk-mode (only virtual console is visible and the application is locked in operate mode)

Parameters: None

Examples:
Start the application in kiosk mode:

```
qlcplus -k
qlcplus --kiosk
```

-l or --locale

Description: Use the given locale for translation

Parameters: Locale name (currently supported: en_GB, fi_FI, fr_FR, es_ES)

Examples:
Use finnish language:

```
qlcplus -l fi_FI
qlcplus --locale fi_FI
```

-o or --open

Description: Open the given workspace file

Parameters: File name

Examples:
Open mydesk.qxw:

```
qlcplus -o mydesk.qxw
qlcplus --open mydesk.qxw
```

-p or --operate

Description: Start the application in Operate mode.

Parameters: None

Examples:
Start QLC+ in operate mode:

```
qlcplus -p
qlcplus --operate
```

-v or --version

Description: Display the current application version number

Parameters: None

Examples:

```
qlcplus -v
qlcplus --version
```

-w or --web

Description: EXPERIMENTAL: Enable remote web access - starts webserver on port 9999

Parameters: None

Examples:

```
qlcplus -w
qlcplus --web
```

Kiosk Mode

QLC+ has a special mode when you need to run the light by inexperienced people, or you just want to prevent unwanted changes. It's called the Kiosk mode. In this mode, only Virtual Console is displayed, so no editing is possible.

When you want to start QLC+ in kiosk mode, run:

```
qlcplus -k -f -o workspace.qxw -p -c 500,10,32,32
```

- **-k** enables the kiosk mode
- **-f** puts QLC+ in fullscreen mode
- **-o workspace.qxw** opens specified workspace
- **-p** starts operate mode
- **-c 500,10,32,32** creates a close button with size of 32x32 pixels, 500 pixels from left and 10 pixels from top.

You can of course specify your own workspace file for **-o**, and size/position for **-c**.

You can omit any of the paramteres if you need so - for example, omit **-c 500,10,32,32** if you don't want close button, and omit **-f** if you don't want fullscreen.

All command-line parameters are described here.

Parameters tuning

This section explains how to manually reach and tune some QLC+ parameters not reachable from the UI.
Keep in mind that if you cannot change them from the UI it means there is a good reason.

Warning: DO NOT edit the configuration files manually unless you know what you're doing. Any misplaced change can cause program crashes or awkward instability.

1 QLC+ configuration location

Linux

Configuration files are located in your user $HOME directory, in the .config/qlcplus folder.
Here's the quick command to access it from a terminal:

```
cd $HOME/.config/qlcplus
```

You will find both QLC+ and Fixture editor configuration files.

Windows

Configuration parameters are stored in the Windows registry.
To access it, run the "regedit" tool and search for the key named "qlcplus".

Mac OSX

Configuration files are located in your user $HOME directory, in the Library/Preferences folder.
Here's the quick command to access it from a terminal:

```
cd $HOME/Library/Preferences
```

QLC+ configuration file is called net.sf.Q Light Controller Plus.plist while the Fixture editor configuration file is called net.sf.Fixture Definition Editor.plist.

2 Configuration Reset

Sometimes it might be necessary to reset QLC+ configuration and bring QLC+ to a "Factory defaults" state.
To do so, locate the configuration as explained in the first paragraph, then do the following:

- On Linux and OSX use the rm 'filename' command to delete the configuration file
- On Windows delete the whole 'qlcplus' KEY using regedit

3 Parameters syntax

Due to Qt diversity on different platforms, parameters are stored in different ways depending on your Operating System.

Linux

Parameters are stored in a plain text file that you can modify with a plain text editor like gedit, kwrite, kate, nano, vim or similar. They are presented as follows:

```
[category]
name=value
```

Windows

Parameters are stored in the Windows registry and can be modified with the regedit tool included in every Windows version. They are presented as follows:

"category" is displayed as a folder
"parameter" is contained in "category" and is represented as a key. Most likely all QLC+ keys are an integer type.
"value" is the actual value of "parameter"

Mac OSX

Parameters are stored in a plain text file that you can modify with a plain text editor like TextWrangler or similar. They are presented as follows:

```
<key>category/name</key>
<string>value</string>
```

4 Parameters list

Most likely you will not find the following parameters in a standard QLC+ configuration. To make them effective, you need to **add** them at the end of your configuration file.

4.1 Simple Desk

Category: simpledesk
Name: channelsperpage
Type: integer
Description: Set the number of DMX channel sliders displayed per page

Category: simpledesk
Name: playbacksperpage
Type: integer
Description: set the number of playback sliders displayed per page

4.2 DMX USB Enttec Open frequency

Category: enttecdmxusbopen
Name: frequency
Type: integer
Description: Set the DMX frame frequency for Enttec Open (and like) devices

4.3 uDMX frequency

Category: udmx
Name: frequency
Type: integer
Description: Set the DMX frame frequency for uDMX devices

4.4 Master Timer frequency

Category: mastertimer
Name: frequency
Type: integer
Description: Set the QLC+ core timer frequency in Hertz (default is 50)

GUI style customization

Starting from version 4.5.0, QLC+ can read a user custom file to change the GUI appearance in a very accessible way.
If no file is found, QLC+ will start with the default style.

The GUI style file

The file name is hardcoded into QLC+ and must be: qlcplusStyle.qss
The style file must also be placed in a specific path which is:

- **Linux**: $HOME/.qlcplus
- **Windows**: Your user folder/QLC+
- **OSX**: Library/Application Support/QLC+

The style file must have a CSS syntax. If you're comfortable with web designing, you should find the creation of this file very easy !
Since the style file is strictly related to the inner Qt objects, you might want to read the following articles to find out the elements names and the additional CSS properties the Qt team added to the default CSS syntax.
Qt Style Sheets
Qt Style Sheets Examples
It is up to your fantasy how you prefer to customize the QLC+ GUI appearance ! If you find a style worth to be shared, don't forget to give your contribution by posting it online in the QLC+ forum

QLC+ dark blue style example

Just to give you an example of how easy this process is, here's a blue-ish dark style for QLC+.
If you copy the following lines into qlcplusStyle.qss in the right location as explained above, you will see it immediately applied to QLC+.

```
QMainWindow, QDialog
{
 background-color: #404B57;
 color: #E6E6E6;
}

QTreeWidget
{
 background-color: #3A444F;
 alternate-background-color: #404B57;
 color: #E6E6E6;
}

QTextBrowser
{
 background-color: #3A444F;
 color: #E6E6E6;
}
```